On Effective Dynamic Search in Specialized Domains

Felipe Moraes
CS Dept, UFMG
Belo Horizonte, MG, Brazil
felipemoraes@dcc.ufmg.br

Rodrygo L. T. Santos
CS Dept, UFMG
Belo Horizonte, MG, Brazil
rodrygo@dcc.ufmg.br

Nivio Ziviani
CS Dept, UFMG & Kunumi
Belo Horizonte, MG, Brazil
nivio@dcc.ufmg.br

ABSTRACT

Dynamic search in specialized domains is a challenging task, in which systems must learn about the user's need from his or her interactive exploration. Despite recent initiatives to advance the state-of-the-art for this task, limited progress has been achieved, with the best performing dynamic search systems only marginally improving upon vanilla ad-hoc search systems. In this paper, we perform a comprehensive analysis of the impact of several components of a prototypical dynamic search system on the effectiveness of the entire system. Through a series of simulations, we discuss the impact of: producing an initial ranking of candidate documents, modeling the possible aspects underlying the user's query given his or her feedback, leveraging the modeled aspects to improve the initial ranking, and deciding when to stop the interactive process. Our results using data from the TREC 2015-2016 Dynamic Domain track shed light on these components and provide directions for the design of effective dynamic search systems for specialized domains.

CCS CONCEPTS

•**Information systems** →**Retrieval effectiveness;** *Users and interactive retrieval;*

KEYWORDS

Dynamic Search; Interactive Search; Search Effectiveness

1 INTRODUCTION

The need for exploration commonly arises in professional search settings such as in medical, legal, patent, military intelligence, and academic search, but also in personal searches such as in travel planning or personal health research [18, 29]. Exploratory searches often involve complex sessions, demanding multiple interactions between the user and a search system. Along the interactive process, the system must dynamically adapt to each feedback provided by the user in order to improve the understanding of the user's need and the usefulness of the subsequently retrieved documents [28].

Research on exploratory search has been supported by several initiatives. The Text REtrieval Conference (TREC) have hosted related research tracks on interactive search [1], search within sessions [6], search for task completion [31] and, more recently, dynamic search in specialized domains [9, 10]. The latter problem, embodied by the TREC Dynamic Domain track,[1] is the focus of this paper.[2] Given an initial query, a dynamic search system must improve its understanding of the user's information need through a series of interactions. In each interaction, the user may provide the system with feedback on the relevance of specific passages of the retrieved documents with respect to one or more aspects underlying his or her information need. The system must then choose to either provide the user with further documents or end the interactive process. An effective system should be able to satisfy as many query aspects as possible (to maximize user satisfaction) with as few interactions as possible (to minimize user effort).

A dynamic search system must cope with four key problems: (i) produce an initial sample of candidate documents given the user's query and the domain of interest; (ii) decide whether the user's information need has been satisfied and eventually stop the interactive process; (iii) leverage the user's feedback to learn an improved aspect model; (iv) produce an enhanced ranking given the learned aspect model. As we will discuss in Section 2, several attempts have been made to produce dynamic search systems that could effectively tackle these problems. Nevertheless, as shown in Figures 1a-b for the two domains considered in the TREC 2016 Dynamic Domain track,[3] even the reportedly most effective system in each domain shows only marginal improvements compared to vanilla ad-hoc search baselines, which leverage no user feedback.

In this paper, we aim to better understand the challenges involved in building effective dynamic search systems. To this end, we isolate each of the aforementioned problems as a separate component of a dynamic search system. Through controlled simulations, we assess how the effectiveness of each component impacts the effectiveness of the whole system. In particular, we show that high-precision document samples are beneficial at early interactions, whereas high-recall samples help towards later interactions. Moreover, mishandled user feedback leads to inaccurate aspect models, which hinder the system effectiveness. Likewise, inaccurately estimating the coverage of the modeled aspects leads to poor reranking, which also hinders effectiveness. Lastly, despite the inherent trade-off, we show that stopping late typically incurs more effort than gain. To our knowledge, this is the first systematic attempt to shed light on the effectiveness of dynamic search in specialized domains.

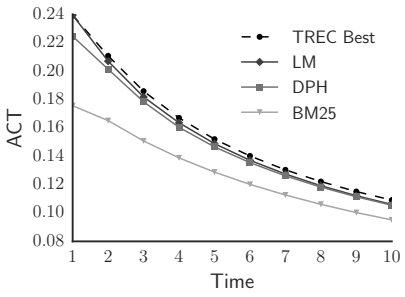
2 RELATED WORK

In this section, we describe relevant related work on exploratory search in general and on dynamic search in particular.

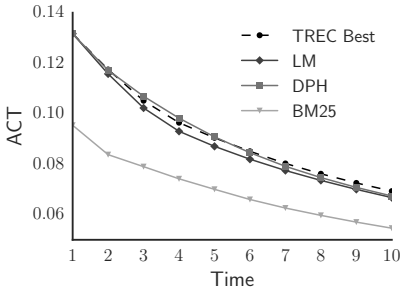
ICTIR'17, October 1–4, 2017, Amsterdam, The Netherlands.

DOI: http://dx.doi.org/10.1145/3121050.3121065

[1] http://trec-dd.org/
[2] A further related task on Dynamic Search for Complex Tasks has recently been proposed at CLEF: https://ekanou.github.io/dynamicsearch/
[3] Each plot shows average cube test (ACT) figures—the primary evaluation metric of the TREC 2016 Dynamic Domain track—along the interactive process. TREC 2015 results follow similar trends and are omitted for brevity.



(a) Ebola 2016



(b) Polar

Figure 1: Per-domain ACT of the best dynamic search systems at the TREC 2016 Dynamic Domain track against ad-hoc search baselines (LM, DPH, and BM25). The best TREC systems were chosen as the most stable across iterations.

2.1 Exploratory Search

Several information retrieval researchers have investigated what makes a search process exploratory in nature [3, 18, 29]. For instance, Marchionini et al. [18] categorized information seeking tasks as lookup (or known-item) search tasks and exploratory search tasks, with the latter being further decomposed into learning and investigation tasks. Another example is the user study conducted by Wildemuth et al. [29], which provided an ample characterization of search tasks. Later, Athukorala et al. [3] described distinctive behaviors of search users during an exploratory search with respect to query length, scroll depth, and task completion time.

Many studies in exploratory search focused on developing user interfaces to support complex information needs [8, 22]. For instance, Ruotsalo et al. [22] proposed an interactive user interface that enhances a user's capacity to explore the results through a visualization of the possible aspects underlying his or her information need. Recently, Krishnamurthy et al. [12] presented an exploratory search system for domain discovery and used the TREC 2015 Dynamic Domain track data to perform user studies. While most research on exploratory search has focused on the user's perspective of the task, here we focus on the effectiveness of exploratory search from a system's perspective. In particular, we address a specific exploratory search task, namely, dynamic search.

2.2 Dynamic Search

Dynamic search is an exploratory search task [30]. Previous research in this area have focused on approaches for session search or multi-page search through reinforcement learning [11, 14, 15, 17, 26]. Sloan et al. [26] proposed a theoretical framework for multi-page dynamic search that learns the best policy based on implicit user feedback, such as clicks. Similarly, Luo et al. [14, 15, 17] proposed several approaches to leverage users' implicit feedback in the form of clicks and query reformulations within a session. These include reinforcement learning approaches such as Markov decision processes, direct policy learning, and dual-agent learning.

In contrast to the aforementioned approaches, we tackle dynamic search to aid user exploration in specialized domains, typically resulting from a focused crawl of the Web. In this setting, users provide explicit feedback on the relevance of each retrieved document with respect to multiple aspects underlying their information need. As part of the evaluation campaigns of the TREC 2015-2016 Dynamic Domain tracks [9, 10], several dynamic search approaches have been proposed that attempt to leverage such a structured feedback. In common, these approaches deploy a multi-step framework for dynamic search, as illustrated in Figure 2.

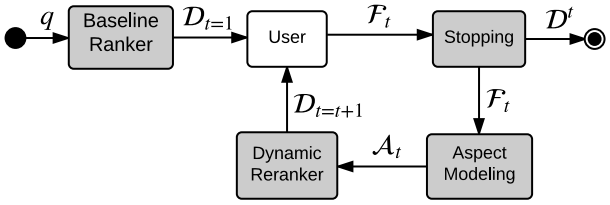


Figure 2: Flow diagram of a typical dynamic search system.

Given the user's query q, at the first step, a *baseline ranker* produces a sample of candidate documents $\mathcal{R}$. Standard ad-hoc retrieval models have been used for this step, including vector space, best matching, and language models [4]. At each time t, the user is presented with a batch of documents $\mathcal{D}_t$ selected from the list $\mathcal{R}$ of candidates and provides a set of feedback $\mathcal{F}_t$. In the second step, a *stopping* mechanism must choose to either continue the interactive process or to end the search session immediately. Stopping heuristics have been proposed that take into account the amount of irrelevant documents observed continuously or cumulatively up to the point of decision [19]. In the third step, the user feedback $\mathcal{F}_t$ on passages extracted from the presented ranking is used to update the system's knowledge about the multiple aspects $\mathcal{A}_t$ underlying his or her information need. Attempted solutions for *aspect modeling* include bag-of-passages modeling [20], clustering [13], and query expansion [21]. Lastly, in the fourth step, the updated aspect model $\mathcal{A}_t$ is used by a *dynamic reranker* to produce an improved result set $\mathcal{D}_{t+1}$, which is again presented to the user for feedback. Effective solutions here include mixture models [32], relevance feedback models [21], and result diversification models [25].

Table 1 organizes the official submissions to the TREC 2015-2016 Dynamic Domain tracks within the general framework described in Figure 2. As discussed in Section 1, not even the best among these approaches was able to consistently improve upon the baseline ranker component alone, which deploys feedback-ignorant ad-hoc search models. The primary goal of this paper is to investigate why this is the case. In the next section, we describe the experimental setup that supports our investigations in Section 4.

Table 1: Overview of dynamic search systems submitted to the TREC 2015-2016 Dynamic Domain tracks.

Group	Year	Baseline Ranker	Stopping	Aspect Modeling	Dynamic Reranker
georgetown_ir	2015	LM	none; cumul.	Passage Relevance	Mixture Models
LavalIVA	2015	Solr def.	none; cont.	Topic Modeling; K-means	Relevance Feedback
uogTr	2015	TF-IDF	none; cumul.	Topic Modeling	Explicit Diversification; Resource Allocation
georgetown	2016	LM; LM+Topic Modeling	none	Query Expansion	Relevance Model
IAPLab	2016	Indri def.+Topic Modeling	none		Markov Decision Processes
LavalLakehead	2016	TFIDF;BM25	none	Topic Modeling; K-Means; Entities	
RMIT	2016	LM	none		Relevance Feedback; Passage Retrieval
ufmg	2016	LM	cumul.; cont.	Passage Relevance	Explicit Diversification
UPD_IA	2016	BM25	none		Quantum Models

3 EXPERIMENTAL SETUP

In this section, we describe the setup that enables our controlled assessment of the effectiveness of dynamic search systems in Section 4. In particular, we detail the test collections and evaluation metrics used in our experiments. Additionally, we describe the reference models used to instantiate the baseline ranker, aspect modeling, and dynamic reranker components in Figure 2.

3.1 Test Collections

Our analysis follows the experimentation paradigm provided by the TREC 2015-2016 Dynamic Domain tracks [9, 10]. The TREC 2015-2016 Dynamic Domain tracks provide test collections targeting the following domains: (i) Ebola, related to the Ebola outbreak in Africa in 2014-2015; (ii) Illicit Goods, related to how illicit and counterfeit goods such as fake Viagra are made, advertised, and sold on the Internet; (iii) Local Politics, related to regional politics, the small-town politicians and personalities in the Pacific Northwest; (iv) Polar, related to the polar sciences. Each domain is indexed separately. In particular, we use Apache Lucene[4] for both indexing and retrieval, with Porter stemmer and standard stopwords removal. For parsing heterogeneous document formats (HTML, XML, RSS, etc.), we use the *AutoDetectParser* of Apache Tika.[5] Moreover, we remove duplicate documents based on their MD5 signature. Salient statistics of all domains are presented in Table 2.

Table 2: TREC 2015-2016 Dynamic Domain track collections. Q is the number of queries. A is the average number of aspects per query. RQ and RA are the average number of relevant documents per query and per aspect. D and U are the number of documents before and after duplicate removal.

Domain	TREC	Q	A	RQ	RA	D	U
Ebola	2015	40	5.7	603	136	6,831,397	5,409,275
Local Politics	2015	48	5.5	141	42	526,717	526,357
Illicit Goods	2015	30	5.3	39	9	497,362	319,538
Ebola	2016	27	4.4	414	121	194,481	193,310
Polar	2016	26	4.7	163	36	244,536	223,141

3.2 Evaluation Metrics

Given the interactive nature of the dynamic search task, we assess the effectiveness of a dynamic search system at different points in time, based upon the batch of documents $\mathcal{D}_t$ presented to the user at each time t. Following standard practice at the TREC 2015-2016 Dynamic Domain tracks [9, 10], we report the average cube test [16][6] at time t (ACT@t). ACT quantifies the ability of a system to fill an "information cube" representing the multiple aspects associated with the user's need. Accordingly, ACT measures the trade-off between the gain attained by satisfying different aspects and the effort incurred in doing so as time goes by. It is defined as:

$$ACT(q, \mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{t} \frac{Gain(q, \mathcal{D}^t)}{Time(\mathcal{D}^t)}, \quad (1)$$

where $Time(\mathcal{D}^t)$ is the amount of time spent examining all documents $\mathcal{D}^t$ presented to the user from the beginning of the interactive process up to time t, and $Gain(q, \mathcal{D}^t)$ is defined as:

$$Gain(q, \mathcal{D}^t) = \sum_{j=1}^{|\mathcal{D}^t|} \sum_{i=1}^{|\mathcal{A}|} \Gamma_{ij}\, \theta_i\, \tilde{g}(a_i, d_j) \mathbb{1}\left(\sum_{k=1}^{j-1} \tilde{g}(a_i, d_k) < M \right), \quad (2)$$

where $\Gamma_{ij} = \gamma^{nrels(a_i, j-1)}$ is a discount factor for novelty, where $nrels(a_i, j-1)$ is the number of relevant documents for aspect a_i among the previously examined documents and $\gamma = 0.5$; θ_i is the a priori importance of aspect a_i, assumed uniformly distributed; $\tilde{g}(a, d)$ is the relevance grade assigned to document d with respect to aspect a, further normalized to lie in the range $[0, 1]$; $M = 1$ is a constant representing the maximum "height" to which the cube for any aspect can be filled; and $\mathbb{1}()$ is the indicator function.

3.3 Reference Components

The baseline ranker component in Figure 2 is responsible for returning a sample of candidate documents $\mathcal{R}$ for a query q. At time $t = 1$, the user is presented with a batch of documents $\mathcal{D}_1$, comprising the five highest scored documents in $\mathcal{R}$. At all other times $t > 1$, $\mathcal{D}_t$ is chosen by the dynamic reranker component from $\mathcal{R} \setminus \mathcal{D}^{t-1}$, where $\mathcal{D}^{t-1}$ denotes all documents returned before time t.

To study the impact of the baseline ranker component, we generate a variety of candidate samples $\mathcal{R}$ by perturbing a reference ranking produced by a field-based weighting model. In particular, we use a field-based extension of DPH [2] (henceforth "DPHF"), a hypergeometric model from the divergence from randomness framework, with field weights set to 0.15 and 0.85 for title and content. Besides being parameter-free, DPHF outperformed similar field-based extensions of best-matching and language models in

[4] https://lucene.apache.org/
[5] https://tika.apache.org/
[6] http://bit.ly/2pq0jq5

our preliminary investigations. Using DPHF, we retrieve the top 1,000 documents as a candidate set $\mathcal{R}$ for each query q.

At any time t, a feedback $f \in \mathcal{F}_t$ for a document $d \in \mathcal{D}_t$ is a tuple $f = \langle a, p, g \rangle$ comprising a passage p that the user deemed relevant to aspect a at a given relevance level $g \in \{1, 2, 3, 4\}$. The structured nature of the feedback associated with each aspect naturally lends itself amenable to some form of aggregate modeling. As illustrated in the magnified portion of Figure 3, as a reference aspect modeling component, we represent each aspect a as an aggregate of the relevant passages associated with it, with the content of each passage p weighted by its corresponding relevance grade g. As time progresses, new passages may be appended to a given aspect tree, and entirely new aspects may be discovered. Moreover, the relative importance of each aspect as perceived by the user can potentially change, as also illustrated in Figure 3 in different shades of gray. To restrict the number of confounding variables in our simulations, we assume a uniform and unchanged aspect importance at all times.

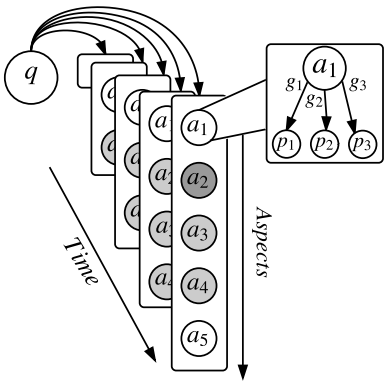


Figure 3: Query aspect modeling over time.

As reference models for dynamic reranking, we take two state-of-the-art diversification models: xQuAD [23] and PM2 [7]. In line with the goal of dynamic search, these models attempt to satisfy as many query aspects as possible (promoting diversity) and as early as possible (promoting novelty) [25].[7] Central to these models are estimates $P(d|q, a)$ of how well document d covers each aspect $a \in \mathcal{A}_t$ of query q. To assess the impact of the dynamic reranker component, we simulate coverage estimates of various performances. In our simulations, we gradually introduce noise into a perfect coverage matrix, which is defined as follows:

$$P(d|q, a) = \frac{g(a, d)}{\sum_{a_i \in \mathcal{A}_t} g(a_i, d)}, \tag{3}$$

where $g(a, d) = \max_{p \in d} g(a, p)$ is the highest relevance grade assigned to any passage p in document d that was judged relevant to aspect a. In other words, when estimating the aspect coverage of a document based on the user's passage-level feedback, we assume that the document is as relevant as its best passage.

4 EXPERIMENTAL EVALUATION

This section analyzes the impact of the four components illustrated in Figure 2 on the effectiveness of a dynamic search system. In particular, we aim to answer the following research questions:

[7]Both xQuAD and PM2 are instantiated with their default setting of $\lambda = 0.5$.

- $Q1$. How does the initial document sample impact the effectiveness of a dynamic search system?
- $Q2$. What is the impact of feedback modeling on the system's knowledge of the aspects underlying the user's query?
- $Q3$. How do improved coverage estimates impact the system's ability to dynamically adapt its ranking strategy?
- $Q4$. What is the impact of early and late stopping strategies on the attained gain-effort trade-off?

The remainder of this section addresses each of these questions in turn. Our observations are based on ACT figures averaged across 171 queries from all five domains summarized in Table 2. Per-domain results follow similar trends and are omitted for brevity.

4.1 Baseline Ranker

The baseline ranker component may impact the effectiveness of a dynamic search system in different moments. In particular, to address *Q1*, we propose two complementary hypotheses:

- $H1$. At earlier interactions, the effectiveness of the system is influenced by the precision attained by the baseline ranker.
- $H2$. At later interactions, the effectiveness of the system is influenced by the recall attained by the baseline ranker.

Regarding *H1*, because little feedback is available at early interactions (with absolutely no feedback at $t = 1$), the overall system effectiveness depends on the relevance of the documents surfaced by the baseline ranker itself (i.e., precision). Regarding *H2*, the potential improvement brought by dynamically reranking the set of candidate documents at later interactions depends on the amount of relevant documents (i.e., recall) available in this set. To test these hypotheses, we simulate baseline rankers of various quality levels. Following Turpin and Scholer [27], for each query q, we generate a series of permutations of the reference ranking $\mathcal{R}$ produced by DPHF (as discussed in Section 3.3) by repeatedly swapping randomly chosen pairs involving one relevant document and one irrelevant document each, until a target average precision (AP) value is achieved. As target AP values for this simulation, we split the range [0,1] of possible values into 20 equally sized bins (i.e., each bin has size 0.05) and randomly select 20 values from each bin, providing a total of 400 simulated permutations per query.

Figure 4 shows the effectiveness of three dynamic search systems (DPHF without reranking, DPHF+xQuAD, and DPHF+PM2) as we vary the quality of the baseline ranking produced by DPHF. Dynamic search effectiveness is given by ACT@t with $t \in \{1, 2, 10\}$,[8] whereas the effectiveness of the baseline ranker is given by either Precision@5 (Figures 4a-c) or Recall@500 (Figures 4d-f). To make sure documents below the 500 cutoff cannot contribute to the reported ACT figures, for this particular experiment, both xQuAD and PM2 are restricted to diversify the top 500 documents returned by the baseline ranker. In addition, to further isolate any impact from the dynamic reranker component itself, both xQuAD and PM2 leverage perfect coverage estimates, as given by Equation (3).

From Figure 4, we first note that dynamic search effectiveness (measured by ACT@t) is highly correlated with the effectiveness of the baseline ranker component (measured by either Precision@5

[8]Note that, at $t = 1$, because no reranking is performed, the rankings produced by all three dynamic search systems (DPHF, DPHF+xQuAD, and DPHF+PM2) are identical.

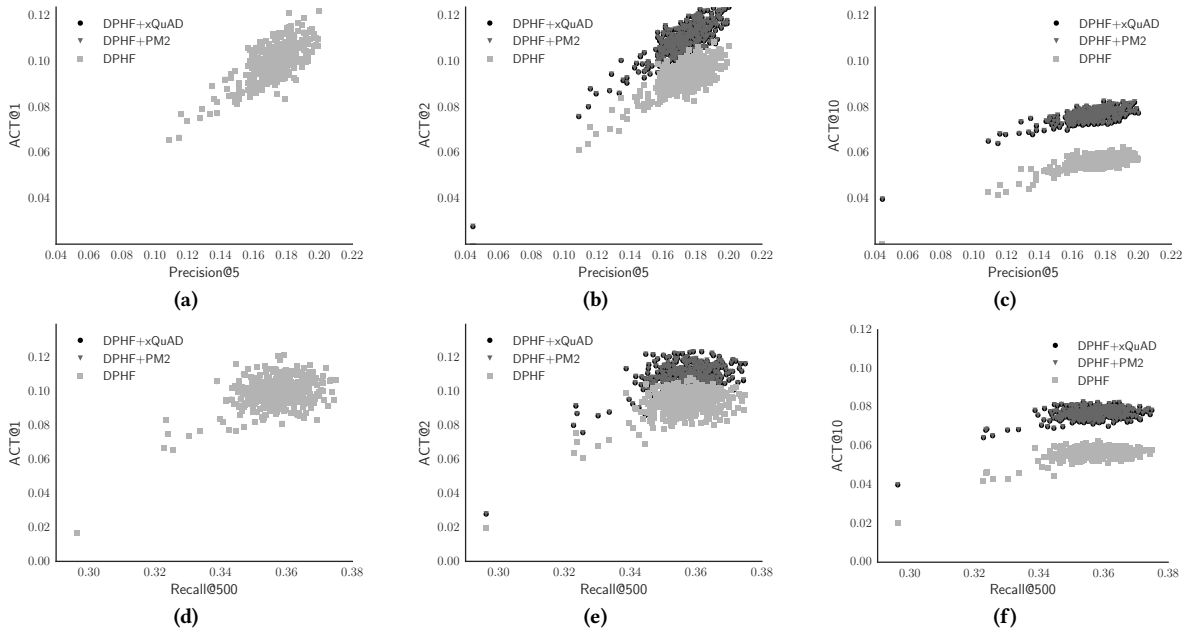


Figure 4: Impact of baseline rankers of various quality levels at times $t \in \{1, 2, 10\}$.

or Recall@500). From Figures 4a-c, we further observe that correlations with Precision@5 are stronger towards early interactions. In contrast, from Figures 4d-f, we note that correlations with Recall@500 are stronger at the last interaction. These observations are further corroborated by the quantitative figures reported in Table 3 in terms of the Pearson correlation between ACT@t and each of these metrics for all considered systems (i.e., DPHF, DPHF+xQuAD, and DPHF+PM2) and all times ($t \in \{1, 2, \ldots, 10\}$). In particular, correlations between ACT@t and Precision@5 peak at time $t = 3$ for DPHF and at $t = 2$ for both DPHF+xQuAD and DPHF+PM2 and then steadily decrease as time progresses. In turn, correlations between ACT@t and Recall@500 steadily increase as time goes by, peaking at $t = 10$ for all three systems. These observations hold regardless of whether the documents returned by the baseline ranker (DPHF) are dynamically reranked (by either xQuAD or PM2) and provide supporting evidence for both *H1* and *H2*. Recalling *Q1*, the experiments in this section demonstrate that an effective baseline ranker impacts the effectiveness of a dynamic search system in different moments, with high-precision baseline rankers improving dynamic search effectiveness at early interactions, and high-recall baseline rankers bringing improvements towards later interactions.

4.2 Aspect Modeling

Section 4.1 showed how the precision and recall of the baseline ranker component may impact the effectiveness of the entire dynamic search system. In this section, we analyze the contribution of an accurate modeling of the multiple aspects $\mathcal{A}_t$ underlying the user's need based upon the feedback $\mathcal{F}_t$ provided by the user at each time t. To address *Q2*, we propose the following hypothesis:

H3. The effectiveness of a dynamic search system can be hindered by an inaccurate or incomplete aspect modeling.

Table 3: Correlation between ACT@t and Precision@5 or Recall@500 attained by the baseline ranker component.

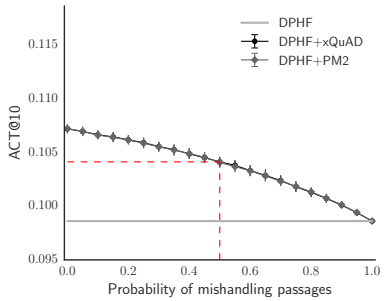
	Precision@5			Recall@500		
t	DPHF	xQuAD	PM2	DPHF	xQuAD	PM2
1	0.8492	0.8492	0.8492	0.5218	0.5218	0.5218
2	0.8760	**0.8837**	**0.8838**	0.5673	0.5576	0.5586
3	**0.8769**	0.8784	0.8781	0.5905	0.5752	0.5764
4	0.8749	0.8686	0.8682	0.6051	0.5835	0.5844
5	0.8713	0.8584	0.8580	0.6143	0.5894	0.5900
6	0.8679	0.8496	0.8493	0.6211	0.5938	0.5943
7	0.8647	0.8421	0.8417	0.6260	0.5968	0.5972
8	0.8621	0.8360	0.8357	0.6300	0.5998	0.6003
9	0.8598	0.8311	0.8307	0.6332	0.6030	0.6035
10	0.8578	0.8274	0.8271	**0.6360**	**0.6061**	**0.6067**

To investigate this hypothesis, we perform two simulations that perturb the reference aspect model described in Section 3.3. First, we simulate the case where we may mishandle some of the user's feedback on different passages associated with a given query aspect a. Let $\kappa_a = \sum_{p \in \cup_t \mathcal{F}_t} \tilde{g}(a, p)$ denote the accuracy of the aspect model built for a, where $\tilde{g}(a, p)$ denotes the relevance grade assigned to a passage p with respect to aspect a, normalized by the total grade of all passages relevant to a (i.e., the "relevance mass" of passage p).[9] In our simulation, a mishandled feedback on passage p for aspect a incurs a probability $(1 - \kappa_a)$ of zeroing out coverage estimates $P(d|q, a)$ of any document d given this aspect, hence introducing noise in the subsequent dynamic reranking. Conversely, with probability κ_a, perfect estimates are used, as defined

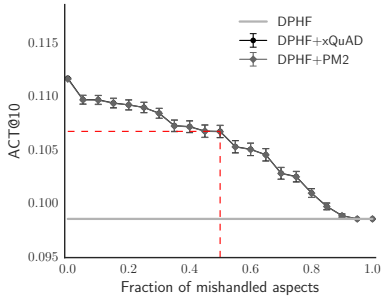
[9] In practice, because the summation encompasses only passages to which the user provided feedback at some time t (as opposed to all relevant passages in the ground-truth), the maximum accuracy an aspect can attain is typically under 1.

in Equation (3). In our second simulation, we consider all aspects as perfectly accurate (i.e., $\kappa_a = 1, \forall a \in \mathcal{A}_t$) and evaluate the impact of incomplete aspect models, by mishandling entire aspects as opposed to individual passages. For this simulation, we zero out coverage estimates $P(d|q,a)$ of all documents that are relevant to a mishandled aspect a.

Figure 5 shows the impact on dynamic search effectiveness in terms of ACT@10 for DPHF, DPHF+xQuAD, and DPHF+PM2 as we perturb the underlying aspect model. To this end, in Figure 5a, we vary the *probability* of mishandling individual passages within the range [0,1] with steps of 0.05. In Figure 5b, we vary the *fraction* of mishandled aspects, also within the range [0,1] with steps of 0.05. In both figures, for a given step, the whole process is repeated 100 times, with error bars denoting standard deviations. Recall that DPHF alone is not affected by perturbations, as it does not leverage any user feedback for reranking. As a result, it provides a natural lower bound for both DPHF+xQuAD and DPHF+PM2.



(a) Inaccurate aspect modeling.



(b) Incomplete aspect modeling.

Figure 5: Impact of inaccurate or incomplete aspect models.

From Figures 5a-b, we note that, as we increase either the probability of mishandling feedback on individual passages or the fraction of mishandled aspects, dynamic search effectiveness is hindered, which answers *Q2* by providing supporting evidence for *H3*. On the other hand, these results demonstrate a reasonable resilience of both xQuAD and PM2 to inaccurate or incomplete aspect models. In particular, as highlighted in Figure 5a, mishandling feedback on individual passages with 50% probability accounts for 35.8% of the total drop in ACT@10. In turn, mishandling 50% of all aspects underlying a query accounts for 37.8% of the total drop in Figure 5b.

4.3 Dynamic Reranker

In Section 4.2, we investigated how perturbed aspect models could impact the effectiveness of a dynamic search system. In that investigation, we isolated the impact of the dynamic reranker component, by leveraging perfect estimates of the coverage of each document with respect to each modeled aspect. In this section, we address *Q3*, by investigating the impact of the dynamic reranker component itself. To this end, we propose the following hypothesis:

H4. The effectiveness of a dynamic search system can be enhanced by improved document coverage estimates for a given aspect model, more so for narrower queries.

Accurate coverage estimates have been shown to contribute to the effectiveness of explicit diversification approaches, such as xQuAD and PM2 [24], which are used here as reference models for dynamic reranking. Our hypothesis is that such estimates will also be key in a dynamic search scenario, particularly for narrower queries, which have a smaller number of relevant aspects and hence are arguably harder to diversify. To test this hypothesis, we simulate increasingly inaccurate coverage estimates, by gradually adding noise to the perfect estimates given by Equation (3). Inspired by related research on differentially private recommender systems [5], we perturb the relevance grade $g(a,d)$ assigned to document d with respect to aspect a by adding a Laplacian noise $Y \sim \text{Laplace}(0,b)$ to it. In this paper, we parameterize b as Δ_a/ϵ. The sensitivity parameter Δ_a captures the dispersion of relevance grades associated with aspect a, as the difference between the maximum and minimum values returned by $g(a,d)$ for all documents $d \in \mathcal{R}$ sampled for the query q. In turn, the leakage parameter ϵ determines how much of the perfect coverage estimates is allowed to "leak" to the dynamic reranker. In other words, lower ϵ values denote noisier coverage estimates, whereas higher ϵ values denote cleaner estimates.

Figure 6 shows the ACT@10 attained by DPHF, DHF+xQuAD, and DPHF+PM2 as we vary the leakage parameter ϵ in the range [0.1,20] with steps of 0.1. For a given step, the entire process is repeated 100 times. On the x-axis, instead of reporting actual leakage values, which cannot be easily interpreted, we indicate how much the resulting (perturbed) coverage estimates differ from perfect coverage estimates. To this end, for each aspect a, we compute the ordering over all documents $d \in \mathcal{R}$ induced by the perturbed coverage estimates $P(d|q,a)$ and compute its nDCG using the expected ordering (induced by the perfect coverage estimates) as ground-truth. The aspect nDCG for a query q is then computed by averaging over the nDCG obtained for its aspects $a \in \mathcal{A}_t$.

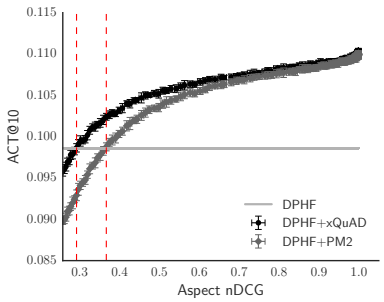


Figure 6: Impact of perturbed coverage estimates.

From Figure 6, we first note that DHF+xQuAD and DPHF+PM2 increasingly outperform the DPHF baseline ranker as their underlying coverage estimates improve, in support of *H4*. In particular, xQuAD begins to outperform DPHF at a critical leakage (CL) point of 0.3, measured in terms of aspect nDCG. On the other hand, PM2 requires slightly improved coverage estimates at a CL point of 0.4. To better understand the impact of improved coverage estimates, Figure 7 provides a breakdown analysis of the results in Figure 6 for queries organized in different bins in the range {1, 2, ..., 10} according to the number of relevant aspects underlying each query.[10]

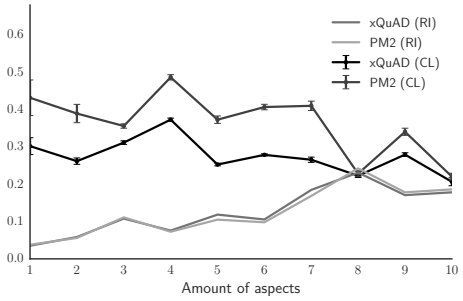


Figure 7: Critical leakage (CL) and room for improvement (RI) for queries with different numbers of relevant aspects.

From Figure 7, we observe a slight decrease in CL as the number of aspects per query increases, particularly for PM2. In addition to CL points, Figure 7 also shows the room for improvement (RI) in each bin, measured as the difference between the ACT@10 attained by the dynamic reranker (xQuAD or PM2) with perfect coverage estimates and the ACT@10 attained by the baseline ranker (DPHF). As shown in the figure, RI increases with the number of aspects per query. These results suggest that narrower queries (i.e., those with fewer aspects) are indeed harder to improve and demand better coverage estimates, which provides further support for *H4*. Recalling *Q3*, the results in this section demonstrate the positive impact of improved coverage estimates—and hence, of an improved dynamic reranker—on the effectiveness of a dynamic search system.

4.4 Stopping Strategies

In the previous sections, we evaluated the impact of the baseline ranker, aspect modeling, and dynamic reranking components under the assumption that the user would interact with a dynamic search system indefinitely. In this section, we address question *Q4*, by investigating the impact of alternative strategies for stopping the interactive process. To this end, we define an oracle stopping strategy, which stops immediately after the last relevant document has been returned, hence providing an optimal gain to the user. At the same time, this strategy may naturally incur additional user effort by extending the interactive process. To better understand this gain-effort trade-off, we simulate suboptimal stopping strategies by increasingly perturbing the stopping decision made by the oracle. Precisely, after receiving the user feedback $\mathcal{F}_t$ at time t, the oracle decides whether or not to stop. With probability τ, this decision is kept; conversely, with probability $(1 - \tau)$, it is flipped.

[10] Queries with more than 10 relevant aspects are discarded from this analysis, as they are substantially fewer, amounting to only 9% of all queries.

Figure 8 shows the dynamic search effectiveness attained by DPHF[11] in terms of ACT@10 as we vary the probability τ of keeping the oracle's stopping decision in the range [0,1] with steps of 0.05. For a given step, the entire process is repeated 100 times, with error bars denoting standard deviations. In addition to the simulated strategies, we consider three heuristic strategies: (i) *none*, which always decides not to stop; (ii) *cumul.*, which decides to stop after observing n_1 irrelevant documents cumulatively; and (iii) *cont.*, which decides to stop after observing n_2 irrelevant documents contiguously. The latter two heuristics were investigated by Maxwell et al. [19] and are tested here with parameters $n_1, n_2 \in \{10, 20\}$.

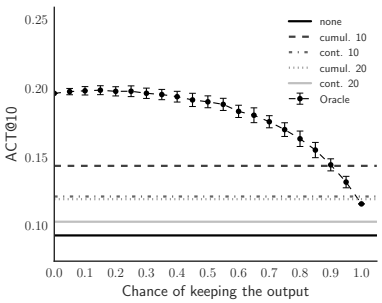


Figure 8: Impact of different stopping strategies for DPHF.

From Figure 8, we first note that the unperturbed oracle strategy (i.e., $\tau = 1$) attains a suboptimal gain-effectiveness trade-off, as measured by ACT@10. While it performs better than the *none* and *cont.* 20 strategies, it is outperformed by all other heuristics, with *cumul.* 10 achieving the highest ACT@10 among them. The underperformance of the oracle is further exacerbated when contrasting it to the increasingly perturbed simulated strategies, which attain the highest ACT@10 by completely flipping all decisions made by the oracle at full perturbation (i.e., $\tau = 0$). This apparent contradiction can be explained by the fact that the oracle strategy optimizes solely for gain, regardless of the incurred effort. In practice, such a gain-oriented strategy tends to stop much later than other effort-oriented strategies. For instance, intuitively, *cumul.* tends to stop earlier than *cont.* since, by definition, $n_1 \leq n_2$ (i.e., it is easier to observe a certain number of irrelevant results cumulatively than contiguously). On the other hand, it is not as apparent why early stopping strategies attain a better gain-effort trade-off. To further analyze this point, we propose the following hypothesis:

H5. Stopping late tends to incur more effort than gain.

To test this hypothesis, we further contrast the five heuristic strategies in Figure 8 in terms of their gain-effort trade-off. Figure 9 breaks down the impact of these heuristics by deconstructing the ACT metric (see Equation (1)) in terms of its two core components: *Gain* and *Time*, with the latter providing a simple proxy for user effort. We observe that strategies that cause no stopping (*none*) or a late stopping (e.g., *cont.* 20) naturally attain more gain compared to early stopping strategies (e.g., *cumul.* 10). Conversely, late stopping strategies naturally incur more effort. We can also observe that, because the amount of relevant documents is finite and typically small, gain rapidly tails off, while effort increases

[11] The same conclusions apply to DPHF+xQuAD and DPHF+PM2.

linearly as time progresses. As a result, although ACT measures the speed of fulfilling the user's information need with multiple aspects, our analysis suggests that it is biased in favor of systems that stop as early as possible, which supports *H5*. Recalling *Q4*, the results in this section demonstrate the trade-off between the gain attained and the effort incurred by continuing the interaction process. While the ACT metric aims at quantifying this trade-off, in practice, the harsh penalty incurred by its effort model discourages late stopping. While alternative effort models could be deployed (e.g., log-based), we leave this investigation to future work.

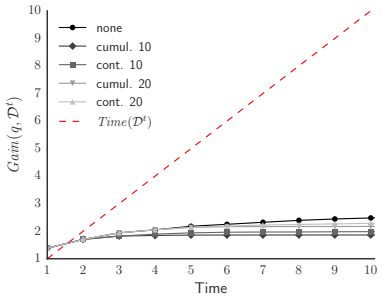


Figure 9: Gain-effort trade-off of stopping strategies.

5 CONCLUSIONS

In this paper, we investigated the role of different components on the effectiveness of a dynamic search system for specialized domains. Through a comprehensive analysis, we found that a high-precision baseline ranker may improve dynamic search at early interactions, whereas a high-recall baseline ranker tends to favor later interactions. Moreover, mishandling the user's feedback on individual passages associated with an aspect or on entire aspects may lead to decreased effectiveness. Likewise, we demonstrated the need for accurately estimating the coverage of each retrieved document with respect to each query aspect, particularly for queries with fewer aspects, which seem inherently harder to improve. Finally, we found that early stopping strategies achieve a better gain-effort trade-off compared to late stopping strategies, which highlights the challenge of promoting effective exploration in this task.

In the future, we plan to extend our analysis to encompass other elements of user interaction, including query reformulations and other forms of implicit feedback, as well as temporal characteristics of the feedback provided within a session. We also plan to further invest in concrete instantiations of each of the four components simulated in this study, so as to harness the potential they demonstrated. In addition to interactive diversification models, a promising direction here includes online learning models. Lastly, another interesting direction for future work involves improved models of the gain-effort trade-off for dynamic search evaluation.

ACKNOWLEDGMENTS

This work was partially funded by projects InWeb (MCT/CNPq 573871/2008-6) and MASWeb (FAPEMIG/PRONEX APQ-01400-14), and by the authors' individual grants from CNPq and FAPEMIG.

REFERENCES

[1] James Allan. 2006. HARD Track Overview in TREC 2005: High Accuracy Retrieval from Documents. In *Proceedings of TREC*.
[2] Giambattista Amati, Edgardo Ambrosi, Marco Bianchi, Carlo Gaibisso, and Giorgio Gambosi. 2016. FUB, IASI-CNR and University of Tor Vergata at TREC 2007 Blog Track. In *Proc. of TREC*.
[3] Kumaripaba Athukorala, Dorota Gffowacka, Giulio Jacucci, Antti Oulasvirta, and Jilles Vreeken. 2016. Is exploratory search different? A comparison of information search behavior for exploratory and lookup tasks. *JASIST* (2016).
[4] Ricardo A. Baeza-Yates and Berthier A. Ribeiro-Neto. 2011. *Modern Information Retrieval - the concepts and technology behind search, Second edition.* Pearson Education Ltd., Harlow, England.
[5] Arnaud Berlioz, Arik Friedman, Mohamed Ali Kaafar, Roksana Boreli, and Shlomo Berkovsky. 2015. Applying Differential Privacy to Matrix Factorization. In *Proc. of RecSys*.
[6] Ben Carterette, Paul Clough, Mark Hall, Evangelos Kanoulas, and Mark Sanderson. 2016. Evaluating Retrieval over Sessions: The TREC Session Track 2011-2014. In *Proc. of SIGIR*.
[7] Van Dang and W. Bruce Croft. 2012. Diversity by Proportionality: An Election-based Approach to Search Result Diversification. In *Proc. of SIGIR*.
[8] Gene Golovchinsky, Abdigani Diriye, and Tony Dunnigan. 2012. The Future is in the Past: Designing for Exploratory Search. In *Proc. of IIIX*.
[9] Grace Hui Yang, John Frank, and Ian Soboroff. 2015. TREC 2015 Dynamic Domain Track Overview. In *Proc. of TREC*.
[10] Grace Hui Yang and Ian Soboroff. 2016. TREC 2016 Dynamic Domain Track Overview. In *Proc. of TREC*.
[11] Xiaoran Jin, Marc Sloan, and Jun Wang. 2013. Interactive Exploratory Search for Multi Page Search Results. In *Proc. of WWW*.
[12] Yamuna Krishnamurthy, Kien Pham, Aécio Santos, and Juliana Freire. 2016. Interactive Exploration for Domain Discovery on the Web. In *Proc. of KDD IDEA*.
[13] S. Lloyd. 2006. Least Squares Quantization in PCM. *IEEE Trans. Inf. Theor.* (2006).
[14] Jiyun Luo, Xuchu Dong, and Hui Yang. 2015. Learning to Reinforce Search Effectiveness. In *Proc. of ICTIR*.
[15] Jiyun Luo, Xuchu Dong, and Hui Yang. 2015. Session Search by Direct Policy Learning. In *Proc. of ICTIR*.
[16] Jiyun Luo, Christopher Wing, Hui Yang, and Marti Hearst. 2013. The Water Filling Model and the Cube Test: Multi-dimensional Evaluation for Professional Search. In *Proc. of CIKM*.
[17] Jiyun Luo, Sicong Zhang, and Hui Yang. 2014. Win-win Search: Dual-agent Stochastic Game in Session Search. In *Proc. of SIGIR*.
[18] Gary Marchionini. 2006. Exploratory Search: From Finding to Understanding. *Commun. ACM* (2006).
[19] David Maxwell, Leif Azzopardi, Kalervo Järvelin, and Heikki Keskustalo. 2015. Searching and Stopping: An Analysis of Stopping Rules and Strategies. In *Proc. of CIKM*.
[20] Felipe Moraes, Rodrygo L. T. Santos, and Nivio Ziviani. 2016. UFMG at the TREC 2016 Dynamic Domain track. In *Proc. of TREC*.
[21] J. J. Rocchio. 1971. Relevance feedback in information retrieval. In *The Smart retrieval system - experiments in automatic document processing*, G. Salton (Ed.). Englewood Cliffs, NJ: Prentice-Hall, 313–323.
[22] Tuukka Ruotsalo, Jaakko Peltonen, Manuel J. A. Eugster, Dorota Glowacka, Aki Reijonen, Giulio Jacucci, Petri Myllymäki, and Samuel Kaski. 2015. SciNet: Interactive Intent Modeling for Information Discovery. In *Proc. of SIGIR*.
[23] Rodrygo L.T. Santos, Craig Macdonald, and Iadh Ounis. 2010. Exploiting Query Reformulations for Web Search Result Diversification. In *Proc. of WWW*.
[24] Rodrygo L. T. Santos, Craig Macdonald, and Iadh Ounis. 2012. On the role of novelty for search result diversification. *Information Retrieval* (2012).
[25] Rodrygo L. T. Santos, Craig Macdonald, and Iadh Ounis. 2015. Search result diversification. *Found. Trends Inf. Retr.* 9, 1 (2015), 1–90.
[26] Marc Sloan and Jun Wang. 2015. Dynamic Information Retrieval: Theoretical Framework and Application. In *Proc. of ICTIR*.
[27] Andrew Turpin and Falk Scholer. 2006. User Performance Versus Precision Measures for Simple Search Tasks. In *Proc. of SIGIR*.
[28] Ryen W. White. 2016. *Interactions with Search Systems.* Cambridge University Press.
[29] Barbara M. Wildemuth and Luanne Freund. 2012. Assigning Search Tasks Designed to Elicit Exploratory Search Behaviors. In *Proc. of HCIR*.
[30] Hui Yang, Marc Sloan, and Jun Wang. 2015. Dynamic Information Retrieval Modeling. In *Proc. of WSDM*.
[31] Emine Yilmaz, Manisha Verma, Rishabh Mehrotra, Evangelos Kanoulas, Ben Carterette, and Nick Craswell. 2015. Overview of the TREC 2015 Tasks Track. In *Proc. of TREC*.
[32] Chengxiang Zhai and John Lafferty. 2001. Model-based Feedback in the Language Modeling Approach to Information Retrieval. In *Proc. of CIKM*.