# ON EFFECTIVE DYNAMIC SEARCH SYSTEMS

FELIPE MORAES GOMES

# ON EFFECTIVE DYNAMIC SEARCH SYSTEMS

Dissertation presented to the Graduate Program in Computer Science of the Federal University of Minas Gerais in partial fulfillment of the requirements for the degree of Master in Computer Science.

Advisor: Rodrygo Luis Teodoro Santos
Co-Advisor: Nivio Ziviani

Belo Horizonte
April 2017

**Ficha catalográfica elaborada pela Biblioteca do ICEx - UFMG**

Gomes, Felipe Moraes.

G633o On effective dynamic search systems / Felipe Moraes Gomes – Belo Horizonte, 2017.
xxiii, 74 f.: il.; 29 cm.

Dissertação (mestrado) - Universidade Federal de Minas Gerais – Departamento de Ciência da Computação.

Orientador: Rodrygo Luis Teodoro Santos
Coorientador: Nívio Ziviani

1. Computação – Teses. 2. Recuperação da informação. 3. Sistemas de recuperação da informação – avaliação. I. Orientador. II. Coorientador. III. Título.

CDU 519.6*73 (043)

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

# FOLHA DE APROVAÇÃO

On effective dynamic search systems

**FELIPE MORAES GOMES**

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:

PROF. RODRYGO LUIS TEODORO SANTOS - Orientador
Departamento de Ciência da Computação - UFMG

PROF. NIVIO ZIVIANI - Coorientador
Departamento de Ciência da Computação - UFMG

PROFA. CLAUDIA HAUFF
WIS - Delft University of Technology

PROFA. JUSSARA MARQUES DE ALMEIDA GONÇALVES
Departamento de Ciência da Computação - UFMG

Belo Horizonte, 20 de abril de 2017.

*To my nephew.*

# Acknowledgments

I would like to express my immense gratitude to many people that contributed to this work in different levels.

First and foremost, I would like to thank my advisor Rodrygo Santos. He has been more than a great advisor, and has spent hours chatting with me, answering my e-mails (more than 300!), and shedding light in many of my decisions. I could not express how thankful I am of having the experience of working with Rodrygo. Besides, I consider Rodrygo a great friend, and I hope that we will have the chance to cheer many beers to come, including in our hometown! Of course, I cannot forget my co-advisor, Nivio Ziviani, to whom I am grateful especially in things beyond academia. He was always available to share his wisdom and point me out directions.

I am thankful to all my professors at UFMG that contributed somehow to this work, but I would like to thank two of them in particular. First, to Jussara Almeida, who taught me for two years the basics of research, being kind and receptive. Also, it was because of her that I decided to pursue a career in computer science. Second, to Adriano Veloso who was open to talk and to give pieces of advise about personal and work life.

This paragraph is dedicated to LATIN, the best laboratory at DCC! I am grateful to all my peers, but above all: Arthur Câmara, Alberto Ueda, Bruno Laporais, Jordan Silva, Marlon Dias, Raul Sanchez, Rafael Glater, and Sabir Ribas.

What is life without family and friends? I am extremely grateful to my parents Divina and João, and my sisters, Kellen and Keroleine. Without their love and dedication to help me reach each step in my life I would not be where I am today. I am so thankful to my friends, which tolerate me and listen to all my complaints about life. I would like to thank my friends in my hometown, Divinópolis, because every time I went home they were there to talk and to celebrate many achievements. I am also thankful to the friends that greatly participated in this step of my life: André Harder, Bruna Neuenschwander, Camila Vieira, Luis Eduardo, Mauri Miguel, Marco Túlio, Mariana Arantes, Patrick Prado, and Rafael Almeida.

*"On a given day, a given circumstance, you think you have a limit. And you then go for this limit and you touch this limit, and you think, "Okay, this is the limit." As soon as you touch this limit, something happens and you suddenly can go a little bit further. With your mind power, your determination, your instinct, and the experience as well, you can fly very high."*

(Ayrton Senna)

# 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 dissertation, we perform a comprehensive analysis of the impact of several components of a typical 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 dynamically rerank the initial set of candidates, and deciding when to stop the interactive process. In addition, we provide preliminary results on practical instantiations of the dynamic reranker component via interactive search result diversification approaches. 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.

**Keywords:** Dynamic Search; Search Systems Effectiveness.

# Resumo

Busca dinâmica em domínios especializados é uma tarefa desafiadora, em que sistemas devem aprender sobre a necessidade do usuário através da sua exploração interativa. Apesar das recentes iniciativas para melhorar o estado da arte para essa tarefa, avanços limitados foram alcançados até então, com os melhores sistemas de busca dinâmica conseguindo melhorias marginais em relação aos sistemas de busca ad-hoc. Nesta dissertação, nós realizamos uma análise abrangente do impacto de vários componentes de um sistema de busca dinâmico genérico sobre a eficácia de todo o sistema. Através de uma série de simulações, discutimos o impacto da produção de um ranking inicial de documentos candidatos, da modelagem dos possíveis aspectos subjacentes à consulta do usuário com seu feedback, da utilização dos aspectos modelados para dinamicamente reordenar o ranking inicial de candidatos e da decisão de quando parar o processo interativo. Além disso, apresentamos resultados preliminares de instanciações práticas do componente de reranking dinâmico por meio de abordagens para diversificação interativa de resultados da busca. Nossos resultados usando dados das coleções de teste da TREC 2015-2016 Dynamic Domain track demonstram o impacto de cada componente e apresentam diretrizes para o projeto de sistemas eficazes para busca dinâmica em domínios especializados.

**Palavras-chave:** Busca Dinâmica, Eficácia de Sistemas de Busca.

# List of Figures

# List of Tables

# Contents

# Chapter 1

# Introduction

Presently, people spend a considerable part of their daily life interacting with information systems such as search engines, recommender systems, and social news feeds, either on desktops or mobile devices. In particular, a user interacts with a search system to find facts, to learn, and to help in making informed decisions. At first, the user interacts with the system through a query. A query is often a short text that is the materialization of a user's information need. The user receives a result list of documents from the system and interacts with the system by looking over the documents or by posing new queries.

To provide a result list of documents, a search system uses a model that is optimized to return the documents that best satisfy the user's information need [Baeza-Yates and Ribeiro-Neto, 2011]. At first, a ranking function sorts a collection of documents using the query terms or query aspects. Afterwards, the ranking function may be updated based on the user's interactions and returns a new result list of documents that more accurately fulfills the user's need. Whenever a user interacts with the system, the system interprets his or her interactions as feedback. Essentially, feedbacks are categorized based on user's input nature: implicit feedbacks in which the user's interactions represent, for instance, clicks, time spent viewing a document, scrolling actions, and eye-tracking; explicit feedbacks in which the user's interactions are graded relevance judgments; and pseudo-feedbacks in which user's interactions are simulated by assuming that the top-$k$ documents will be assessed positively by the user.

As information technology grows, users have been using search systems in an exploratory manner to complete tasks that are more complex. 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 [Wildemuth and Freund, 2012; Marchionini,

2006]. Exploratory searches often involve complex search 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 quality of the subsequently retrieved results [White, 2016].

Research on exploratory search has been supported by several initiatives. The Text REtrieval Conference (TREC) has hosted related research tracks on interactive search Allan [2006], search within sessions Carterette et al. [2016], search for task completion Yilmaz et al. [2015] and, more recently, dynamic search in specialized domains Grace Hui Yang [2015, 2016]. 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).

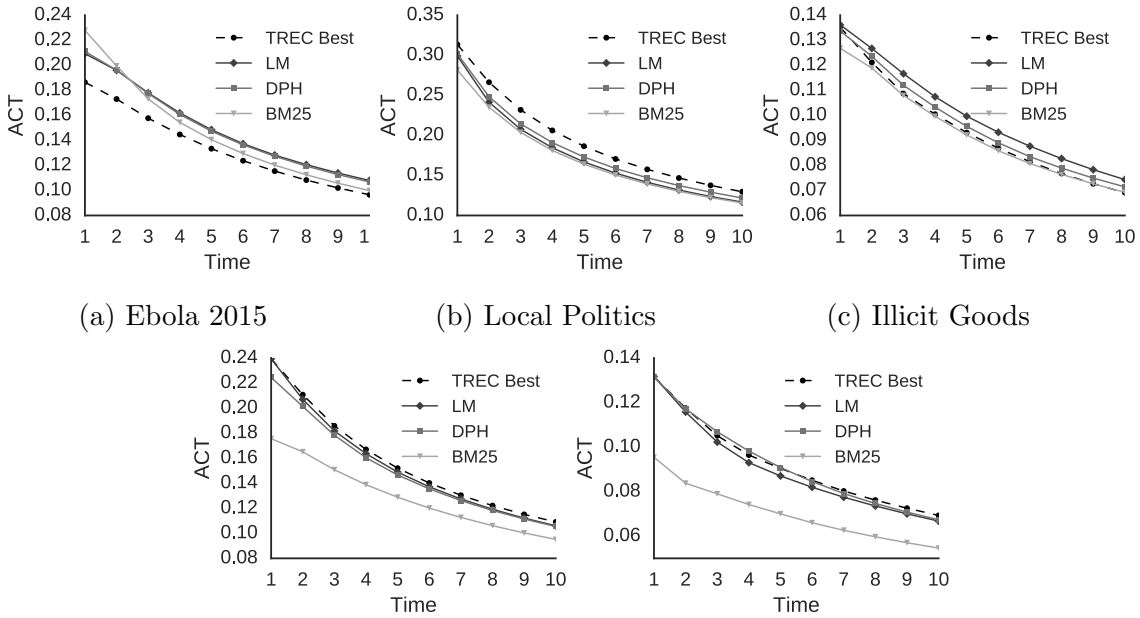
A dynamic search system must cope with four key problems: (i) produce an initial sample of candidate documents given the user's query and the domain of interest; (ii) decide whether the user's information need has been satisfied and eventually stop the interactive process; (iii) leverage the user's feedback to learn an improved aspect model; (iv) produce an enhanced ranking given the learned aspect model. As we will discuss in Chapter 2, several attempts have been made to produce dynamic search systems that could effectively tackle these problems. Nevertheless, as shown in Figures 1(a)-(e) 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 dissertation, 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 effective-

[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.




(a) Ebola 2015 (b) Local Politics (c) Illicit Goods

(d) Ebola 2016 (e) Polar

Figure 1.1: Per-domain ACT of the best dynamic search systems submitted to the TREC 2015-2016 Dynamic Domain track against ad-hoc search baselines (LM, DPH, and BM25). The best TREC systems were chosen as the most stable across iterations.

ness 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. In addition, we present a preliminary analysis of practical instantiations of one component that dynamically reranks the initial ranking of candidate documents. To our knowledge, this is the first systematic attempt to shed light on the effectiveness of dynamic search in specialized domains.

## 1.1 Motivation

Our motivation for this dissertation, besides having observed little improvement in the state-of-the-art for dynamic search systems for specialized domains, stems from the practical importance of such systems due to the explosive growth of text data. This includes Web pages, scientific papers, electronic health records, microblog posts, and social network posts. Such large data collections have generated a new challenge for

practitioners to retrieve relevant information in their day-to-day tasks. There are several examples of these tasks as in health search, legal search, patent search, marketing research, and scientific literature review. For illustration, take the following scenarios:

- Health search: a physician needs to review all the patients' records that satisfy some conditions to select a group for clinical trials;
- Legal search: a lawyer needs to find every piece of evidence related to his or her case from documents that are under legal hold;
- Patent search: before launching a product, a tech company needs to check all the patents related to their product.
- Marketing research: a social analyst needs to identify all the different posts in which a rumor disseminates in order to recreate the diffusion process and measure the impact of the rumor;
- Scientific literature review: scientists need to find all pieces of prior work that are related to their research.

## 1.2 Dissertation Statement

The statement of this dissertation is that the current state-of-the-art in dynamic search in specialized domains is hindered by the suboptimal effectiveness of different dynamic search components, namely, candidate result sampling, user feedback modeling, dynamic reranking, and interaction stopping. In particular, this dissertation aims to answer the following research questions:

*Q1* – How does each component of a dynamic search system impact the overall system effectiveness?

*Q2* – Can a practical instantiation of the framework through interactive diversification improve over a vanilla ad-hoc search baseline?

In order to answer the above research questions, in this dissertation, we explore the task proposed by the TREC Dynamic Domain track. We investigate a working instantiation of a general framework for dynamic search system via interactive search result diversification and evaluate the impact of each component on the effectiveness of the system as a whole. Then, we present a preliminary investigation of the results of using practical instantiations of this framework.

## 1.3 Contributions

Our main contributions are three-fold:

- **Contribution 1:** A thorough analysis of the impact of different components on the effectiveness of a dynamic search system.
- **Contribution 2:** A preliminary investigation on practical instantiations of one component of dynamic search system via interactive search result diversification.

## 1.4 Dissertation Outline

The remainder of this dissertation is structured as follows:

- **Chapter 2: Related Work** presents related work on exploratory search, dynamic search, and search quality. Moreover, it presents a unifying characterization of state-of-the-art dynamic search systems for specialized domains.
- **Chapter 3: Dynamic Search via Interactive Search Result Diversification** presents a working instantiation of the introduced dynamic search framework that will be used in our experiments.
- **Chapter 4: Experimental Setup** presents test collections, evaluation metrics, and vanilla ad-hoc search system baselines for the conducted evaluation.
- **Chapter 5: Simulation Results** investigates the impact of different components on the effectiveness of a dynamic search system.
- **Chapter 6: Practical Instantiations** presents preliminary results on practical instantiations of a dynamic search system via interactive diversification.
- **Chapter 7: Conclusion and Future Works** concludes the dissertation, summarize our findings and proposes a set of future research directions.

# Chapter 2

# Related Work

In this chapter, we describe related work on exploratory search in general in Section 2.1 and on dynamic search in particular in Section 2.2. In addition, we present a general framework for dynamic search systems in Section 2.2. Furthermore, we discuss search result quality and evaluation of dynamic search systems in Section 2.3.

## 2.1 Exploratory Search

Several information retrieval researchers have investigated what makes a search process exploratory in nature [Wildemuth and Freund, 2012; Marchionini, 2006; Athukorala et al., 2016]. For instance, Marchionini [2006] 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 (as shown in Table 2.1). Another example is the user study conducted by Wildemuth and Freund [2012], which provided an ample characterization of search tasks. Later, Athukorala et al. [2016] 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 [Ruotsalo et al., 2014; Golovchinsky et al., 2012]. For instance, Ruotsalo et al. [2014] 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. [2016] 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

| | |
|---|---|
| **Learning** | Knowledge Acquisition<br>Comprehension or interpretation<br>Comparison<br>Aggregation or Integration<br>Socialize |
| **Investigation** | Accretion<br>Analysis<br>Exclusion or Navigation<br>Synthesis<br>Discovery<br>Planning or forecasting<br>Transformation |

Table 2.1: Subtasks of learning and investigation tasks based on Marchionini [2006].

of exploratory search from a system's perspective. In particular, we address a specific exploratory search task, namely, dynamic search.

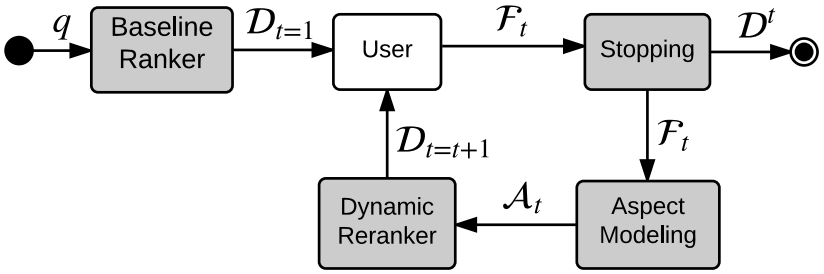



Figure 2.1: Flow diagram of a typical dynamic search system.

## 2.2 Dynamic Search

Dynamic search is an exploratory search task [Yang et al., 2016]. Previous research in this area have focused on approaches for session search or multi-page search through reinforcement learning [Luo et al., 2014, 2015a,b; Sloan and Wang, 2015; Jin et al., 2013]. Sloan and Wang [2015] 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. [2014, 2015a,b] 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

Table 2.2: 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 |

Web [Barbosa and Freire, 2007]. 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 [Grace Hui Yang, 2015, 2016], 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 2.1.

Given the user's query $q$, in the first step, a *baseline ranker* produces an initial result set $\mathcal{R}$, which is presented to the user. Standard ad-hoc retrieval models have been used for this step, including vector space [Salton et al., 1975], best matching [Robertson et al., 1996], and language models [Zhai, 2008]. 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 [Maxwell et al., 2015]. In the third step, the user feedback $\mathcal{F}$ on passages extracted from the presented ranking is used to update the system's knowledge about the multiple aspects $\mathcal{A}$ underlying his or her need. Attempted solutions for *aspect modeling* include direct passage modeling, clustering [Lloyd, 2006], and query expansion [Zhai, 2008; Rocchio, 1971]. Lastly, in the fourth step, the updated aspect model $\mathcal{A}$ is used by a *dynamic reranker* to produce an improved result set $\mathcal{D}$, which is again presented to the user for feedback. Effective solutions here include mixture models [Zhai and Lafferty, 2001], relevance feedback models [Rocchio, 1971], and result diversification models [Santos et al., 2015].

Table 2.2 organizes the official submissions to the TREC 2015-2016 Dynamic Domain tracks within the general framework described in Figure 2.1. As discussed in Chapter 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 dissertation is to investigate why this is the

case. In the next chapter, we introduce a working instantiation of the aforementioned framework to enable a controlled assessment of the impact of all four components illustrated in Figure 2.1 on the performance of a dynamic search system. To our knowledge, this is the first systematic attempt to shed light on the effectiveness of dynamic search systems for specialized domains.

## 2.3 Search Quality

The quality of a search experience is a prerogative of the users of a search system. Therefore, an ideal evaluation would occur by directly enquiring users about their experience. However, such intervention could affect the overall user search experience. Consequently, many researchers have proposed different methodologies for evaluating search effectiveness (see Kanoulas [2016] for a survey).

User studies have been a common evaluation methodology, whereby a selected group of users perform a set of instructed tasks. In such methodology, besides capturing the user interactions with the system, the users answer a questionnaire during and after the experiment. However, a small modification of a search system would require a new user study, which makes this methodology expensive. In light of this, commercial search engines prefer to evaluate *in situ*, for example, using A/B tests with implicit feedback (e.g., clicks, dwell-time, scrolling depth, and others). Although *in situ* evaluation is less expensive than user studies, it is somewhat restricted to the access of commercial search engines. Thus, many researchers have been improving the methodology of collection-based evaluation in many campaigns such as in the Text REtrieval Conference (TREC), and the Conference and Labs of the Evaluation Forum[1] (CLEF). This evaluation methodology has many advantages such as its reusability, being also a less expensive and intrusive approach [Sanderson, 2010].

In a collection-based evaluation, similarly to user studies, assessors (a selected group of users) evaluate their search experience with respect to their satisfaction with a pool of documents for an instructed task. The produced relevance assessments form a benchmark that is used later to evaluate other search systems. To build the test benchmark, we need to cope with how to capture the user satisfaction about a document and how to summarize the user satisfaction with a search system, which we further discuss in the remainder of this section.

[1] http://clef2017.clef-initiative.eu/

### 2.3.1 User Satisfaction

In a TREC-style evaluation, the user satisfaction is simplified to the pragmatic view of relevance judgment. Many factors impact the assessors relevance judgment about a document with the foremost factor being the instructions given to the assessors. In addition, Xu and Chen [2006] investigated other five factors, including topicality, novelty, understandability, reliability, and scope. In particular, of the five factors, they found that topicality and novelty were the two essential ones for relevance judgment on a document.

Several researchers have argued that relevance is not a proper measure for capturing user satisfaction with a search system. Cooper [1997] was the first to argue that utility of a document should be used instead of relevance as a measure of user satisfaction. Moreover, Saracevic [1975, 2007a,b] argued that search systems should return documents not only relevant but also based on cost and benefits as in economy theory. Additionally, in his recent work, Saracevic [2007a,b] suggests that effort should be considered jointly with relevance in the evaluation of search systems.

Furthermore, recent user studies have found mismatches between user satisfaction and relevance judgments of collection-based evaluation. Yilmaz et al. [2014] found that many of these mismatches might be explained by the absence of judgment based on effort. Later, Verma et al. [2016] conducted a user study by questioning users about the effort they spent on finding relevant information within a document. They analyzed three factors that may impact the effort judgment of a user: findability, readability, and understandability. Also, they found that new features should be considered in the design of effort-focused search systems.

Jiang et al. [2017] conducted a user study to capture the user satisfaction within a search system, which they defined as the *ephemeral state of relevance* (ESR). They investigated several factors that impact the ESR, including novelty, effort, understandability, and reliability. In a similar conclusion to that of Verma et al. [2016], Jiang et al. [2017] found that effort along with other factors explains the ESR. In particular, they found that novelty significantly impact the judgment of user satisfaction with a search system. Furthermore, Luo et al. [2017] concluded that the perception of time spent consuming information might be influenced by the relevance of a document.

In this dissertation, we rely upon the relevance perspective proposed by the TREC-style evaluation. However, our approach to summarize the user satisfaction with a search system attempts to simulate the user satisfaction beyond the perspective of relevance judgment of a document.

### 2.3.2 Summarization of User Satisfaction

To summarize the user satisfaction, we use evaluation metrics. Several evaluation metrics have been proposed based on the concept of relevance judgment. Set-based metrics summarize the user satisfaction by extending metrics such as precision, recall, and average precision. Position-based and cascade-based metrics such as Rank-biased Precision (RBP) and Expected Reciprocal Ranking (ERR) model the user's behavior in analyzing the search results [Moffat and Zobel, 2008; Chapelle et al., 2009]. As discussed in Section 2.3.1, several user studies indicated that novelty and effort should be considered to evaluate search system effectiveness. Instead of instructing user's to judge based on these factors, some evaluation metrics attempt to simulate this type of user judgment, for instance, effort-based metrics and novelty and diversity metrics.

Many effort-based metrics have been proposed in the literature. Smucker and Clarke [2012] proposed the Time-Based Gain (TBG), a general framework that takes into account the time spent by a user reading a document, accounting on whether a document is relevant. Later, Clarke and Smucker [2014] introduced an instantiation of TBG, which they called the Time Well Spent (TWS), which considers how users read a trail of documents by simulating the speed at which they read a piece of text. Sakai and Dou [2013] presented U-measure, an even more general framework inspired by TBG. The U-measure metric can be used to evaluate not only search results on documents but also passages of documents and question answering tasks. Recently, Ferro et al. [2016] proposed Twist, a metric that measures the avoidable effort that a user could obtain based on the position of the graded relevance judgment of the search result list. Although these metrics attempt to take into account the effort a user spends consuming relevant information, all of them still rely on the TREC-style of relevance. Despite the formulation account for being general, none of them has been employed to evaluate a complex information need as is common in dynamic search tasks.

In a dynamic search task, information needs are inherently ambiguous or faceted. Specifically, in such tasks, we can decompose a complex information need in several aspects. Therefore, many researchers have proposed novelty-based and diversity-based metrics to evaluate such complex information needs. Most of these metrics adapted the relevance-based metrics to cope with multiple aspects or intents of an information need, which are *intent-aware* metrics (ERR-IA, MAP-IA) [Agrawal et al., 2009; Chapelle et al., 2011]. Additionally, other metrics that are worth mentioning, which also target the novelty and diversity scenario, are the Novelty Relevance-biased Precision (NRBP), and $\alpha-$nDCG (Clarke et al. [2008]). Although these metrics account for complex information needs, it is not clear whether they incorporate the effort of acquiring

information about a document in their formulation.

In this dissertation, we use the *Average Cube Test* (ACT) metric proposed by Luo et al. [2013]. ACT belongs to a family of metrics that measures the search speed of a user in fulfilling a complex information need [Jiang and Allan, 2016]. In particular, ACT explicitly computes the benefit-effort ratio a user has when consuming information. Although ACT is similar to TBG [Smucker and Clarke, 2012] in its basic formulation, it incorporates the capacity of evaluating complex information need regarding diversity and novelty of multiple aspects. For these reasons, we mainly use ACT to evaluate the effectiveness of a dynamic search system.

## 2.4 Summary

In this chapter, we introduced related work on exploratory search and dynamic search. In Section 2.1, we outlined exploratory search tasks and the research conducted to improve such tasks. In Section 2.2, we briefly presented the literature in dynamic search, including session search and multi-page search. Then, we characterized the dynamic search systems submitted to the TREC 2015-2016 Dynamic Domain track and elaborated a unifying framework that highlighted the characteristics of these submitted systems. Last, in Section 2.3, we discussed evaluation methodologies for search systems in general as well as search metrics. In the following chapter, we present a working instantiation of the general framework presented in this chapter.

# Chapter 3

# Dynamic Search via Interactive Search Result Diversification

Dynamic search in a specialized domain brings several challenges. For instance, to find about "hand washing importance" in a domain covering Ebola outbreak in Africa, a dynamic search system should find documents covering multiple aspects of the user's underlying information need, such as "Sierra Leone campaign" and "Nigeria campaign" so as to maximize user satisfaction. At the same time, to minimize user effort, this broad coverage of aspects should be achieved as early as possible. This informal definition resembles that of a related problem, which has received considerable attention from the IR community in recent years: search result diversification [Santos et al., 2015]. However, in contrast to the standard diversification problem, dynamic search is inherently interactive. In particular, a dynamic search system must interact with the user in order to continually identify aspects of interest that should be covered by the subsequently returned documents. Inspired by state-of-the-art diversification approaches [Santos et al., 2010; Dang and Croft, 2012; Hu et al., 2015], in this chapter, we introduce a working instantiation of the general framework for dynamic search described in Figure 2.1, as a means to analyze the impact of each of its components.

## 3.1 Baseline Ranker

In the baseline ranker component, a retrieval model returns a list of candidate documents $\mathcal{R}$ for a query $q$. It also provides a relevance estimate $rel(q, d)$ for each document $d \in \mathcal{R}$ to be used by the next components. Without loss of generality, following the standard evaluation paradigm proposed by the TREC Dynamic Domain track, we consider user interactions performed on batches of five documents each. Precisely, 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 $\mathcal{F}_t$. At time $t = 1$, before any feedback is received, $\mathcal{D}_1$ comprises the five highest scored documents in $\mathcal{R}$. At all other times $t > 1$, $\mathcal{D}_t$ is chosen by the dynamic reranker component (see Section 3.4) from $\mathcal{R} \setminus \mathcal{D}^{t-1}$, where $\mathcal{D}^{t-1}$ (note the superscript) denotes all documents returned before time $t$.

Intuitively, the baseline ranker may impact the effectiveness of the entire system in two distinct moments. At time $t = 1$, because no feedback is available and hence no further reranking is performed, the effectiveness of the system depends on the precision attained by the baseline ranker itself. At all other times $t > 1$, the effectiveness of the system is more influenced by the recall of the baseline ranker. Indeed, with the availability of user feedback starting from $t = 2$ to be leveraged by the dynamic reranker, the attainable effectiveness of the system will be limited by the amount of relevant documents available in the candidate ranking $\mathcal{R}$. In order to assess the impact of the baseline ranker on the entire system, we instantiate this component through a series of simulations, producing candidate rankings $\mathcal{R}$ of a range of different performances.

## 3.2 Stopping

As illustrated in Figure 2.1, at a given time $t$, the user is shown a batch of documents $\mathcal{D}_t$ generated either by the baseline ranker or the dynamic ranker component. At this point, the set of feedback $\mathcal{F}_t$ provided by the user is fed to the second component in the framework: a stopping mechanism. The role of this mechanism is to predict whether the user's information need has been fulfilled or there is more knowledge to be gained through further interactions with the system. Stopping too early may result in reduced user satisfaction, while stopping too late may result in wasted user effort.

To take an informed decision, the stopping mechanism must assess all the feedback provided by the user thus far. Because documents without feedback are considered irrelevant for evaluation purposes, we consider the amount of irrelevant documents observed as the main criterion for a stopping decision. In particular, inspired by related research on user browsing behavior [Maxwell et al., 2015], we instantiate the stopping mechanism with three different strategies:

**none** The interactive process never stops.

**cont.** The interactive process stops after $n_1$ irrelevant documents have been observed continuously.

**cumul.** The interactive process stops after $n_2$ irrelevant documents have been observed cumulatively.

The "none" strategy serves as a baseline for the assessment of the other two. Regarding the parameters of "cont." and "cumul.", Maxwell et al. [2015] originally reported effective $n_1$ values ranging from 1 to 20, and effective $n_2$ values ranging from 25 to 50. Given the multi-batch nature of our interactive setting and the considered batch size of only five documents, we experiment with multiple values for $\alpha$ and $\beta$ in a lower range, as we discuss in Section 5.4. In order to assess the impact of the stopping mechanism on the entire system, we instantiate this component through a series of simulations, producing stopping strategies of a range of different performances.

## 3.3 Aspect Modeling

Whenever the stopping mechanism allows the interactive process to continue, the user feedback $\mathcal{F}_t$ is made available to improve the subsequently returned document batches. In particular, 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 aspect modeling component is responsible for leveraging this feedback in order to continuously improve its knowledge of the possible aspects $\mathcal{A}_t$ underlying the user's information need. While external resources (e.g., query reformulations [Santos et al., 2010], taxonomy classes [Agrawal et al., 2009]) could also be used for an improved modeling, we leave their exploration to future work and focus on aspect modeling based on the user feedback alone.

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 3.1, each aspect $a$ can be directly represented as an aggregate of the relevant passages associated with it, with the content of each passage $p$ possibly 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 3.1 with different shades of gray. In order to assess the impact of this component on the effectiveness of a dynamic search system, we simulate aspect models with various levels of completeness, by randomly removing either passages from different aspects or entire aspects altogether. To restrict the number of confounding variables in our simulation, we assume a uniform and unchanged aspect importance at all times.

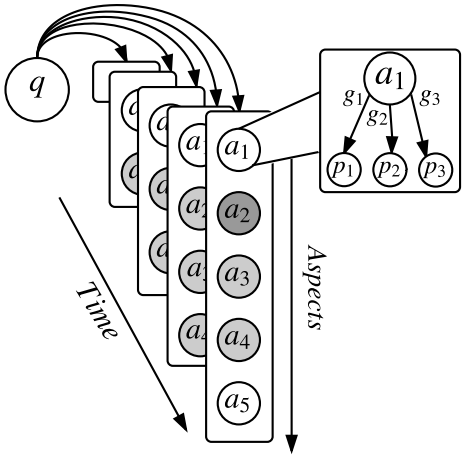



Figure 3.1: Query aspect modeling over time.

## 3.4 Dynamic Reranker

Given the aspect model $\mathcal{A}_t$ and the set of all documents $\mathcal{D}^t$ presented up to time $t$, the dynamic reranker must produce an ideally improved ranking $\mathcal{D}_{t+1} \subseteq \mathcal{R} \setminus \mathcal{D}^t$ of documents from the candidate set $\mathcal{R}$ that are yet to be seen by the user. Several approaches could be used to instantiate this component, including passage-level relevance feedback [Na et al., 2008] or a mixture of passage-specific language models [Li and Zhu, 2008]. However, one important objective of a dynamic search system is to achieve a high coverage of multiple aspects as early as possible, in order to reduce the incurred user effort. As discussed in the beginning of this section, search result diversification models [Santos et al., 2015] are a natural fit in this scenario, as they aim to improve aspect coverage with minimum redundancy in the ranking.

Achieving maximum aspect coverage at a fixed ranking depth has been demonstrated to be an NP-hard problem [Agrawal et al., 2009]. Most diversification approaches in the literature adopt a well-known greedy algorithm to this problem, which approximates the optimum within a factor of $(1 - 1/e) \approx 0.632$ [Nemhauser et al., 1978]. In particular, to build the target ranking $\mathcal{D}_{t+1}$, the greedy algorithm iteratively selects one document $d^* \in \mathcal{R} \setminus \mathcal{D}^t$ at a time according to the following objective:

$$d^* = \arg\max_{d \in \mathcal{R} \setminus \mathcal{D}^t} score(q, d, \mathcal{A}_t, \mathcal{D}^t \cup \mathcal{D}_{t+1}), \tag{3.1}$$

where $score()$ quantifies the extent to which document $d$ covers the set of aspects $\mathcal{A}_t$ as well as how novel the document is in light of the documents already presented to the user. Regarding the latter, we posit that novelty should be estimated with respect to not only the documents previously selected for the target batch $\mathcal{D}_{t+1}$, but also the

ones presented to the user in all previous batches, namely, $\mathcal{D}^t$. This slight modification makes existing diversification approaches directly applicable to the dynamic search setting. In the remainder of this section, we describe four state-of-the-art approaches, which two of which are used in our simulations of the dynamic reranker component. In our practical instantiations, we compare the four diversification approaches listed here.

## 3.4.1 Coverage-based models

### 3.4.1.1 Flat models

Santos et al. [2010] introduced the xQuAD framework, which estimates *score*() in Equation (3.1) as the probability that document $d$ covers explicitly identified aspects $\mathcal{A}$ underlying the query $q$ that are not well covered by the already selected documents in $\mathcal{D}$. Precisely,[1] we have:

$$score_{xQuAD}(q, d, \mathcal{A}, \mathcal{D}) = (1 - \lambda) rel(q, d) + \lambda div(q, d, \mathcal{A}, \mathcal{D}), \tag{3.2}$$

where $rel(q, d)$ is given by the baseline ranker component (see Section 3.1) and $div(q, d, \mathcal{A}, \mathcal{D})$ is defined as:

$$div(q, d, \mathcal{A}, \mathcal{D}) = \sum_{a \in \mathcal{A}} P(a|q)\, P(d|q, a) \prod_{d_j \in \mathcal{D}} (1 - P(d_j|q, a)), \tag{3.3}$$

where $P(a|q)$ denotes the relative importance of aspect $a$ given $q$, $P(d|q, a)$ denotes the coverage of document $d$ with respect to this aspect, and the rightmost product denotes the novelty of any document covering this aspect, according to how badly this aspect is covered by the documents previously observed in $\mathcal{D}$. Importantly, while no modification to the actual xQuAD formulation is needed, its input changes at every timestep $t$. Indeed, as illustrated in Figures 3.1 and 3.2, both the set of aspects $\mathcal{A}_t$ as well as the coverage matrix, comprising coverage estimates $P(d|q, a)$ for every document-aspect pair, are updated at every timestep $t$.

To assess the impact of the dynamic reranker component on the whole system, we simulate coverage estimates of various performances. To this end, in our simulations, we gradually introduce noise into a perfect coverage matrix, which is defined as follows:

$$P(d|q, a) = \frac{g(a, d)}{\sum_{a_i \in \mathcal{A}_t} g(a_i, d)}, \tag{3.4}$$

[1] Subscripts and superscripts have been omitted for readability.

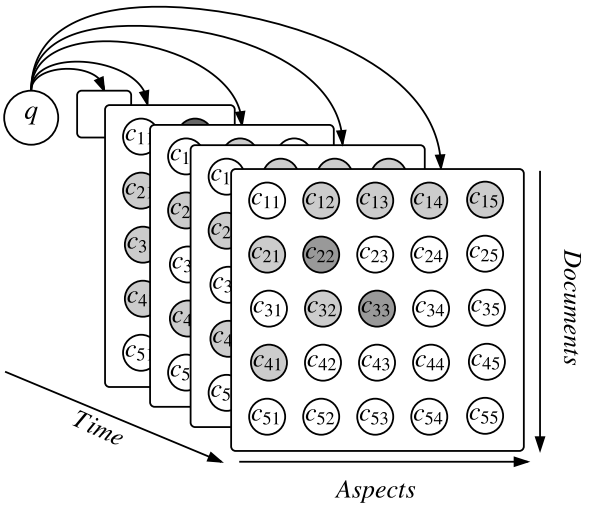



Figure 3.2: Aspect coverage over time.

where $g(a, d)$ is the highest relevance grade assigned to any passage $p$ in document $d$ that was judged relevant to aspect $a$. In other words, we assume that a document is as relevant as its best passage.

#### 3.4.1.2 Hierarchical models

Hu et al. [2015] proposed HxQuAD as an extension of the xQuAD framework to support diversification using hierarchically organized aspects. Such hierarchy can be modeled as a tree in which each node represents an aspect and the set of aspects at the $i$-th level of the tree is denoted $\mathcal{A}_i$. In Figure 3.3, we illustrate a example of aspect tree for a query $q$. Therewith, we can define the diversity of a document $d$ with respect to aspects at the $i$-th level according to:

$$div_i(q, d, \mathcal{D}) = \sum_{a \in \mathcal{A}_i} P(a|q)\, P(d|q, a) \prod_{d_j \in \mathcal{D}} (1 - P(d_j|q, a)), \tag{3.5}$$

where both $P(a|q)$ and $P(d|q, a)$ are defined recursively, so that, for any non-leaf aspect $a$ with children $\mathcal{C}$, we have:

$$P(a|q) = \sum_{c \in \mathcal{C}} P(c|q) \quad \text{and} \quad P(d|q, a) = 1 - \prod_{c \in \mathcal{C}} (1 - P(d|q, c)). \tag{3.6}$$

Given these definitions, HxQuAD estimates the overall diversity of document $d$ by

linearly combining its diversity estimates at multiple hierarchy levels, according to:

$$div_H(q,d,\mathcal{D}) = \alpha div_1(q,d,\mathcal{D}) + (1-\alpha) div_2(q,d,\mathcal{D}) + \frac{(1-\alpha)^2}{\alpha} div_3(q,d,\mathcal{D}) + ... + \frac{(1-\alpha)^{n-1}}{\alpha^{n-2}} div_n(q,d,\mathcal{D}), \quad (3.7)$$

where the $\alpha$ hyperparameter controls the influence of different hierarchy levels in the final estimation, with $\alpha = 0.5$ indicating that all levels are equally weighted. Finally, HxQuAD score function is defined in a similar manner as of xQuAD:

$$score_{HxQuAD}(q,d,\mathcal{A}_H,\mathcal{D}) = (1-\lambda) rel(q,d) + \lambda div_H(q,d,\mathcal{A},\mathcal{D}). \quad (3.8)$$

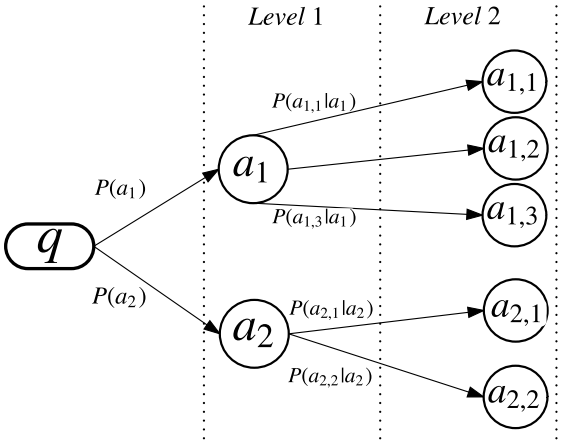



Figure 3.3: Hierarchical aspect tree for a query $q$.

In our practical instantiations, we consider a tree as the aspect $a$ and the passages $p$ which belongs to $a$ as the children $\mathcal{C}$ of $a$.

## 3.4.2 Proportionality-based models

### 3.4.2.1 Flat models

Dang and Croft [2012] presented diversification models aimed to cover query aspects in the ranking proportionally to their popularity. We use PM2, a probabilistic adaptation of the Sainte-Laguë method originally proposed to assign parliament seats to parties after an election. At each iteration, the algorithm first selects the best aspect based on a proportionality strategy and then finds the most relevant document optimized for the selected aspect. Precisely, we have:

$$score_{PM2}(q, d, \mathcal{A}, \mathcal{D}^t) = \lambda \frac{v_*}{2s_* + 1} P(d|q, a_*) + (1 - \lambda) \sum_{a_j \in \mathcal{A} \setminus a_*} \frac{v_j}{2s_j + 1} P(d|q, a_j), \tag{3.9}$$

where $v_j = P(a_j|q)$ and $s_i$ denotes the proportion of seats assigned to $a_i$. A parameter $\lambda$ trades off relatedness to aspect $a_*$ vs. all other aspects. At each document selection, $a_*$ is selected as the highest quotient $v_i/(2s_i + 1)$. After selecting $d^*$ based on Equation (3.9), $s_i$ is update as:

$$s_i = s_i + \frac{P(d|q, s_i)}{\sum_{s_j \in \mathcal{S}} P(d|q, s_j)} \tag{3.10}$$

#### 3.4.2.2 Hierarchical models

Hu et al. [2015] presented HPM2 as an extension of the PM2 framework to support diversification using hierarchically organized aspects. Similarly to HxQuAD, we have:

$$score_{HPM2}(q, d, \mathcal{A}, \mathcal{D}) = \alpha score_1(q, d, \mathcal{A}_1, \mathcal{D}) + (1 - \alpha) score_2(q, d, \mathcal{A}_2, \mathcal{D}) + \frac{(1 - \alpha)^2}{\alpha} score_3(q, d, \mathcal{A}_3, \mathcal{D}) + ... + \frac{(1 - \alpha)^{n-1}}{\alpha^{n-2}} score_n(q, d, \mathcal{A}_n, \mathcal{D}), \tag{3.11}$$

where the $\alpha$ hyperparameter controls the influence of different hierarchy levels in the final estimation, with $\alpha = 0.5$ indicating that all levels are equally weighted. We denote the set of aspects at the $i$-th level of the tree $\mathcal{A}_i$. Next, we defined $score_i$ as:

$$score_i(q, d, \mathcal{A}_i, \mathcal{D}^t) = \lambda \frac{v_*}{2s_* + 1} P(d|q, a_*) + (1 - \lambda) \sum_{a_j \in \mathcal{A}_i \setminus a_*} \frac{v_j}{2s_j + 1} P(d|q, a_j) P(a_j|a_i) \tag{3.12}$$

where both $v_j = P(a_j|q)$ and $P(d|q, a_*)$ are defined recursively as in Equation (3.6). $a_*$ is a selected aspect node at each aspect level $i$ in $\mathcal{A}_i$. A parameter $\lambda$ trades off relatedness to sub-aspect node $a_*$ vs. all other aspects nodes. $P(a_j|a_*)$ is the weight of a aspect $a_j$ based on its distance to the selected aspect node $a_*$, which is defined as:

$$P(a_j|a_*) = \frac{2j - dis(a, a_*)}{2j} \tag{3.13}$$

where $dis(a, a_*)$ is the length of the path for moving from $a$ to $a_*$. As both aspect nodes are at level $j$, the maximal distance between $a$ and $a_*$ is $2j$, which is used to normalize the distance. To select a document $d^*$, we first select an aspect node $a_*$ as the highest quotient $v_i/(2s_i + 1)$. After selecting $d^*$ based on Equation (3.11), $s_i$ is updated as:

$$s_i = s_i + \frac{P(d|q, a_i)}{\sum_{a_j \in \mathcal{A}_)} P(d|q, a_j)} \tag{3.14}$$

## 3.5 Summary

This chapter introduced a working instantiation of the general framework for dynamic search systems presented in Section 2.2. In Section 3.1, we presented the baseline ranker component, which is responsible for providing a set of candidate documents. Section 3.2 presented the stopping component and stopping strategies studied in this dissertation. In Section 3.3, with the feedback received from the user, we showed a component that model the user's feedback and forward it to the dynamic reranker component to generate a new batch of documents to the user. Thus, in Section 3.4, we introduced an instantiation of the component via interactive search result diversification. In particular, we adapted state-of-the-art search result diversification models in order to tackle the dynamism of the user's interaction. In the next chapters, we thoroughly investigate the framework presented in this chapter. In Chapter 5, we investigate the impact of each component of this framework through a series of simulations. Finally, in Chapter 6, we present a preliminary investigation on practical instantiations of the dynamic reranker component using external evidence to estimate the aspect coverage of the dynamic reranker component.

# Chapter 4

# Experimental Setup

In this chapter, we describe the general setup for the experiments conducted in Chapter 5 and 6. In particular, our experiments aim to answer the following research questions:

*Q1* – How does each component of a dynamic search system impact the overall system effectiveness?

- *Q1.1* – How does the initial document sample impacts the effectiveness of a dynamic search system?
- *Q1.2* – What is the impact of feedback modeling on the system's knowledge of the aspects underlying the user's query?
- *Q1.3* – How do improved coverage estimates impact the system's ability to dynamically adapt its ranking strategy?
- *Q1.4* – What is the impact of early and late stopping strategies on the attained gain-effort trade-off?

*Q2* – Can a practical instantiation of the framework through interactive diversification improve over a vanilla ad-hoc search baseline?

- *Q2.1* – Can we use external evidence to improve the aspect coverage estimates?
- *Q2.2* – What queries are improved the most and the least?

In the remainder of this chapter, in Section 4.1, we detail the test collections, whereas Section 4.2 presents the evaluation metrics used in our experiments. In Section 4.3, we describe the ad-hoc search system settings used to return a list of candidate documents in the first step of the framework presented in Chapter 3.

## 4.1 Test Collections

We conduct our effectiveness analysis within the standard experimentation paradigm provided by the task of the TREC 2015-2016 Dynamic Domain tracks [Grace Hui Yang, 2015, 2016]. In particular, this task comprises topics in four domains, as presented in Table 4.1. They are: (i) Ebola, which is related to the Ebola outbreak in Africa in 2014-2015; (ii) Illicit Goods, which is related to how illicit and counterfeit goods such as fake Viagra are made, advertised, and sold in the Internet; (iii) Local Politics, which is related to regional politics, the small-town politicians and personalities in the Pacific Northwest; (iv) Polar, which is related to the polar sciences. To retrieve novel documents as soon as possible, we perform a duplication removal procedure. To this end, we generated an MD5 hash signature for each document and iterated through them and removing the duplicated ones. In Table 4.1, Docs and Deduped are the number of documents before and after the duplicate removal. Moreover, in Table 4.2, we present an example of query topic, sub-topics, and document passages for some of the subtopics. Note that we use the term aspect instead of sub-topic as originally proposed only to facilitate the comparison with the literature.

Table 4.1: TREC 2015-16 Dynamic Domain track collections. T is the number of topics. S is the average number of sub-topics per topic. RT and RS are the average number of relevant documents per topic and per sub-topic, respectively.

| Domain | TREC | T | S | RT | RS | Docs | Deduped |
|---|---|---|---|---|---|---|---|
| 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 |

Table 4.2: TREC 2015 Dynamic Domain track, topic DD15-33, along with its corresponding sub-topics and a few passages.

| **Domain** | Local Politics | **Passages** |
|---|---|---|
| **Topic** | DD15-33 Australian Nursing Home Fire | DD15-33.1<br>4 There were 87 residents in the home on Hambledon Road when the fire broke out.<br>4 AS 96 elderly men and women lay in their beds the flames were building around them.<br>4 There were up to 100 residents in the nursing home and all have now been removed. |
| **Sub-topics** | DD15-33.1 Patients<br>DD15-33.2 Liability<br>DD15-33.3 Cause<br>DD15-33.4 Arrest and Charges<br>DD15-33.5 Fatalities and Injuries | DD15-33.5<br>4 Alma Smith 73 died at the scene of the fire in Quakers Hill on Friday morning.<br>4 Lola Bennett 86 died at Royal North Shore Hospital on Saturday afternoon. |

## 4.2 Evaluation metrics

In this section, we present the evaluation metrics used in our experiments. In Section 4.2.1, we detail the primary metric used in this dissertation, and in Section 4.2.2, we present additional metrics used in this work.

### 4.2.1 Average Cube Test

In this dissertation, we use the *Cube Test* evaluation metric proposed by Luo et al. [2013], which measures the speed of completing a user's need or task $q$ represented as a cube. Luo et al. [2013] defined the average search speed at timestep $t$ as:

$$CT(q,t) = \frac{Gain(q,t)}{Effort(q,t)}, \tag{4.1}$$

where $Gain(q,t)$ is the amount of information the user found and $Effort(q,t)$ is the effort the user spent on the search task. Thus, by using Equation 4.1, Luo et al. [2013] derived the average $CT$ as:

$$ACT(q,t) = \frac{1}{\mathcal{N}}\frac{1}{t}\sum_{k=1}^{t}\frac{Gain(q,k)}{Effort(q,k)}, \tag{4.2}$$

where $\mathcal{N}$ is a normalization factor which maps the $ACT$ score into the range 0–1. Next, we describe *Gain*, and *Effort*.

#### 4.2.1.1 Gain

The amount of information is related to how much an object has been influenced by another object. *Gain* measures the amount of information the user found from the search system. Given a list of documents $\mathcal{D}^t = \{d_1, d_2, d_3, ..., d_t\}$ retrieved by a search system to the user up to timestep $t$, we define *Gain* as:

$$Gain(q,t) = \sum_{i=1}^{t} Gain(q,d_i,t), d_i \in \mathcal{D}^t \tag{4.3}$$

where $Gain(q,d,t)$ is the amount of information a document $d$ conveys to the user at a timestep $t$. In dynamic search tasks, the user's information need can be defined by multiple aspects $\mathcal{A}$ and a document may contain information that is relevant to one or more aspects. To this end, Luo et al. [2013] proposed to break the cube into multiple cuboids, with each cuboid representing a different aspect $a \in \mathcal{A}$. Accordingly,

the amount of information conveyed by a document is defined as:

$$Gain(q, d, t) = \sum_{a \in \mathcal{A}} I(q, a)\, UPR(q, d, a, t) Nov(q, a, t), \tag{4.4}$$

where $I(q, a)$ is the importance of an aspect $a$, which stands for the area of the cuboid representing $a$. $UPR(q, d, a, t)$ is the *user-perceived relevance* of a document $d$ for a query $d$ and aspect $a$, which illustrates the extent to which a document contributes to fulfill the cuboid $a$. Finally, $Nov(q, a, t)$ is the novelty of an aspect $a$, which comes across as the friction faced when fulfilling the aspect cuboid $a$. In this dissertation, we follow the official definitions of the three factors described above that were proposed by the TREC Dynamic Domain organizers: $I(q, a)$ is uniform for all aspects; $Nov(q, a, t)$ is a decay function $\gamma^{na}$, where $\gamma$ is a value ranging between 0–1, and $na$ is the number of documents found about aspect $a$ in the previously selected documents; and $UPR$ is defined as

$$UPR(q, d, a, t) = URR(d, q, a) \mathbb{1} \left( \sum_{k=1}^{t} URR(d_k, q, a) Nov(q, a, t) < T \right), \tag{4.5}$$

where $URR$ (User-Received Relevance) measures the amount of relevant information a document $d$ conveys to a user with respect to an aspect $a$ into the range 0–1. In contrast to $UPR$, $URR$ is independent from the previously selected documents. $UPR_r$ and $UPR_n$ measure $UPR$ by assuming that there is a maximum amount of information a user can absorb about an aspect $a$. Consequently, right after reaching a threshold $T$ there is no absorption of document $d$ by the user.

#### 4.2.1.2 Effort

*Effort* measures the amount of time or cost spent investigating the list of documents until a timestep $t$. In this dissertation, we follow the official definition of *Effort* proposed by the TREC Dynamic Domain organizer: $\lfloor t/b \rfloor$, where $t$ is the number of documents and $b$ the batch size which is five.

### 4.2.2 Novelty and Diversity Metrics

Besides the Average Cube Test, we evaluate our practical instantiations of a dynamic search system with $\alpha$-normalized discounted cumulative gain ($\alpha$-nDCG, [Clarke et al., 2008]), a metric that balances relevance and diversity through the tuning parameter

$\alpha$, defined as:

$$\alpha\text{-nDCG}(q,t) = \frac{1}{\mathcal{N}} \sum_{k=1}^{t} \frac{\sum_{a \in \mathcal{A}} rel(d_k, a)(1-\alpha)^{\sum j=1^{k-1} rel(d_k,a)}}{log_2(k+1)}, \tag{4.6}$$

where $rel(d_k, a)$ is the (binary) relevance grade of the $k$-the ranked document with respect to each aspect $a \in \mathcal{A}$, and $\mathcal{N}$ is the normalization factor from the traditional nDCG. The larger the value of $\alpha$, the more diversity is rewarded. In contrast, when $\alpha = 0$, only relevance is rewarded, and this metric is equivalent to the nDCG. In addition, we use the intent-aware (IA) of the expected reciprocal ranking (ERR) [Chapelle et al., 2011], defined as:

$$\text{ERR-IA}(q,t) = \sum_{a \in \mathcal{A}} P(a|q)\text{ERR}(a,t), \tag{4.7}$$

where $\text{ERR}(q,t)$ is computed individually for each aspect $a \in \mathcal{A}$, following the assumption that none of the other query aspects is of interest of the user.

## 4.3 Ad-hoc Search System

As an ad-hoc search system, we use the open source search library Lucene[1] for both indexing and retrieval, with Porter stemmer and standard English stopwords removal. Additionally, we indexed each data domain separately. For document parsing (HTML, XML, RSS, etc.), we use the *AutoDetectParser* of Apache Tika.[2]

We use a field-based extension of DPH [Amati et al., 2007] (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 our preliminary investigations. Using DPHF, we retrieve the top 1,000 documents as a candidate set $\mathcal{R}$ for each query $q$.

## 4.4 Summary

This chapter introduced the research questions underlying the proposed dissertation and provided details of the general experimental setup used to answer them in the

[1] https://lucene.apache.org
[2] https://tika.apache.org/

following chapters. In Section 4.1, we presented the test collections used in the evaluation of the dynamic search framework discussed in the Chapter 3. In Section 4.2, we presented the primary metric used in this work and introduced the other support metrics used in Chapter 6. In Section 4.3, we presented the ad-hoc search system settings used throughout this work. In the following chapters, we answer the first group of research questions in Chapter 5, and in Chapter 6, we answer the second group of research questions.

# Chapter 5

# Simulation Results

In this chapter, we evaluate the impact of each component of the framework presented in Chapter 3.[1] In particular, our experiments aim to answer the following research questions related to *Q1*, "How does each component of a dynamic search system impact the overall system effectiveness?":

*Q1.1* – How does the initial document sample impacts the effectiveness of a dynamic search system?

*Q1.2* – What is the impact of feedback modeling on the system's knowledge of the aspects underlying the user's query?

*Q1.3* – How do improved coverage estimates impact the system's ability to dynamically adapt its ranking strategy?

*Q1.4* – What is the impact of early and late stopping strategies on the attained gain-effort trade-off?

To this end, we propose hypotheses related to the above research questions and answer the questions in the following sections: Section 5.1 studies the impact on the initial list of documents in the system; Section 5.2 studies the contribution of the user's feedback to update the system's knowledge about the aspects underlying a user's need; Section 5.3 investigates the impact of coverage estimates for dynamically reranking the initial list of documents; Section 5.4 analyzes different strategies for stopping the interactive search process.

[1] To avoid any bias towards any of these components, we applied the standard setting of $\lambda = 0.5$ to balance relevance and diversity in the case of xQuAD, and current best aspect novelty and the novelty of other aspects in the case of PM2.

## 5.1 Baseline Ranker

The baseline ranker component may impact the effectiveness of a dynamic search system in different moments. In particular, to address *Q1.1*, 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 [2006], for each query $q$, we generate a series of permutations of the reference ranking $\mathcal{R}$ produced by DPHF (as discussed in Section 4.3) 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 5.1 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 $t \in \{1, 2, 10\}$,[2] whereas the effectiveness of the baseline ranker is given by either Precision@5 (Figures A.5a-c) or Recall@500 (Figures A.5d-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 (3.4).

From Figure 5.1, we first note that dynamic search effectiveness (measured by ACT@$t$) is highly correlated with the effectiveness of the baseline ranker component

[2]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.

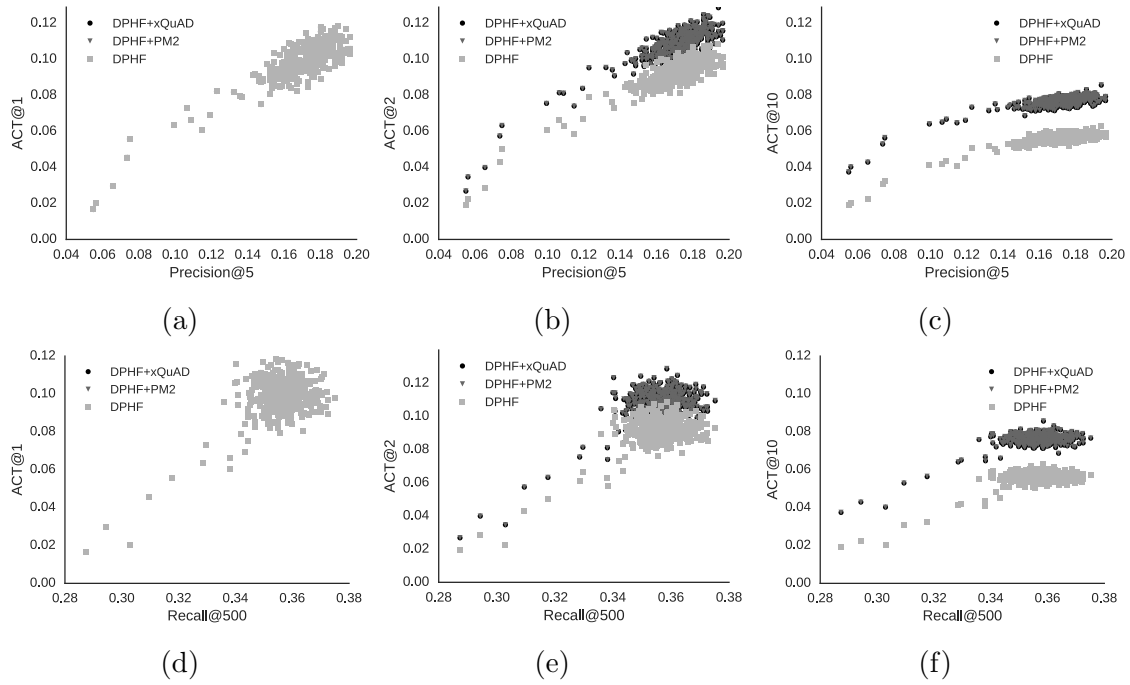



Figure 5.1: Impact of baseline rankers of various quality levels at times $t \in \{1, 2, 10\}$ (see Appendix A.1 for separated domains).

(measured by either Precision@5 or Recall@500). From Figures A.5a-c, we further observe that correlations with Precision@5 are stronger towards early interactions. In contrast, from Figures A.5d-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 5.1 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 ($t \in \{1, 2, ..., 10\}$). 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 *Q1.1*, 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.

Table 5.1: 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** |

## 5.2 Aspect Modeling

Section 5.1 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 *Q1.2*, we propose the following hypothesis:

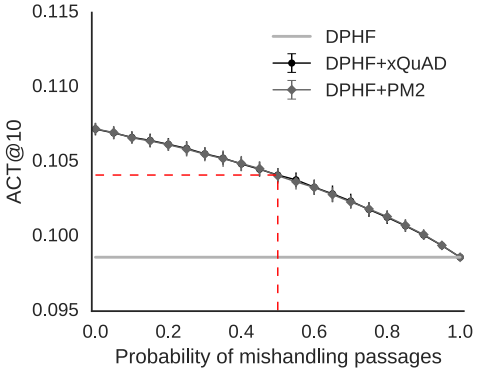
*H3*. The effectiveness of a dynamic search system can be hindered by an inaccurate or incomplete aspect modeling.

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 $p$).[3] 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 $\kappa_a$, perfect estimates are used, as defined in

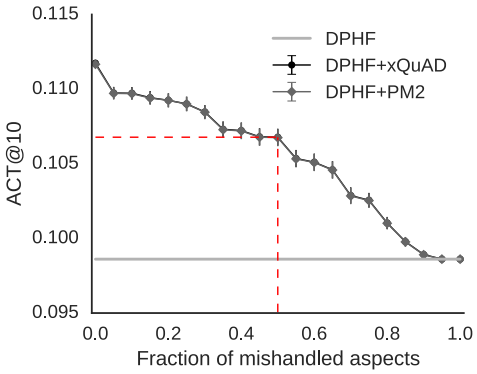
[3]In practice, because the summation encompasses only passages to which the user provided feedback at some time $t$ (as opposed to all relevant passages in the ground-truth), the maximum accuracy an aspect can attain is typically under 1.

Equation (3.4). 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 5.2 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 5.2a, we vary the *probability* of mishandling individual passages within the range [0,1] with steps of 0.05. In Figure 5.2b, 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 5.2: Impact of inaccurate or incomplete aspect models (see Appendix A.2 for separated domains).

From Figures 5.2a-b, we note that, as we increase either the probability of mis-

handling feedback on individual passages or the fraction of mishandled aspects, dynamic search effectiveness is hindered, which answers *Q1.2* 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 5.2a, 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 5.2b.

## 5.3 Dynamic Reranker

In Section 5.2, 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 *Q1.3*, 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 [Santos et al., 2012], 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 innacurate coverage estimates, by gradually adding noise to the perfect estimates given by Equation (3.4). Inspired by related research on differentially private recommender systems [Berlioz et al., 2015], 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 dissertation, we parameterize $b$ as $\Delta_a/\epsilon$. The sensitivity parameter $\Delta_a$ 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 $\epsilon$ determines how much of the perfect coverage estimates is allowed to "leak" to the dynamic reranker. In other words, lower $\epsilon$ values denote noisier coverage estimates, whereas higher $\epsilon$ values denote cleaner estimates.

Figure 5.3 shows the ACT@10 attained by DPHF, DHF+xQuAD, and DPHF+PM2 as we vary the leakage parameter $\epsilon$ 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 $a \in \mathcal{A}_t$.

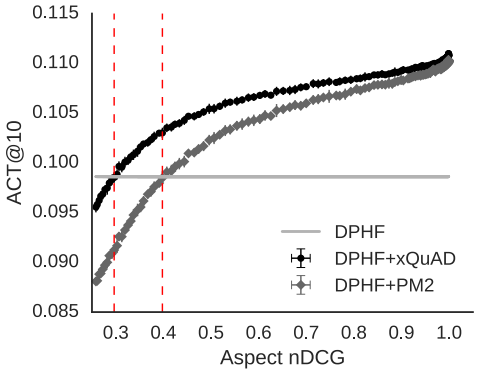



Figure 5.3: Impact of perturbed coverage estimates (see Appendix A.3 for separated domains).

From Figure 5.3, 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 5.4 provides a breakdown analysis of the results in Figure 5.3 for queries organized in different bins in the range $\{1, 2, ..., 10\}$ according to the number of relevant aspects underlying each query.[4]

From Figure 5.4, we observe a slight decrease in CL as the number of aspects per query increases, particularly for PM2. In addition to CL points, Figure 5.4 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

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

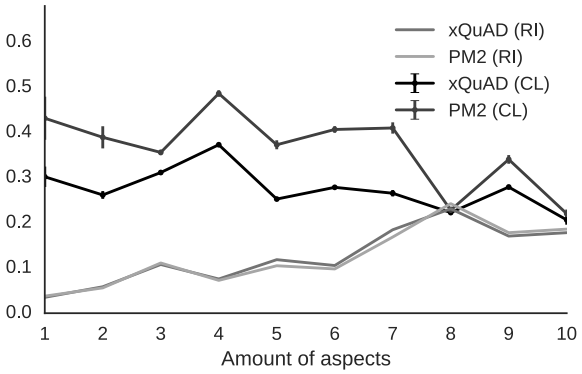



Figure 5.4: Critical leakage (CL) and room for improvement (RI) for queries with different numbers of relevant aspects.

demand better coverage estimates, which provides further support for *H4*. Recalling *Q1.3*, 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.

## 5.4 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 *Q1.4*, 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 $\tau$, this decision is kept; conversely, with probability $(1 - \tau)$, it is flipped.

Figure 5.5 shows the dynamic search effectiveness attained by DPHF[5] in terms of ACT@10 as we vary the probability $\tau$ 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)

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

*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. [2015] and are tested here with parameters $n_1, n_2 \in \{10, 20\}$.

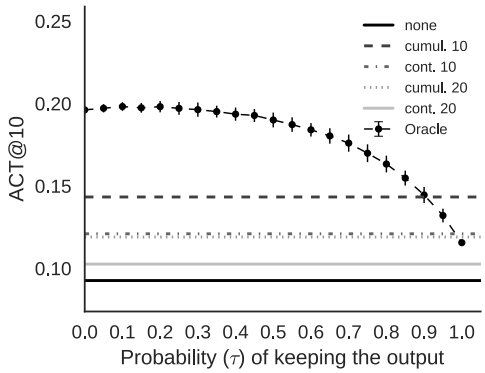



Figure 5.5: Impact of different stopping strategies for DPHF.

From Figure 5.5, 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 5.5 in terms of their gain-effort trade-off. Figure 5.6 breaks down the impact of these heuristics by deconstructing the ACT metric (see Equation (4.2)) 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 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 *Q1.4*, 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.

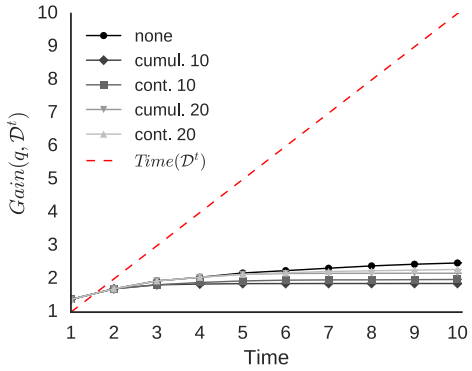



Figure 5.6: Gain-effort trade-off of stopping strategies.

## 5.5 Summary

In this chapter, we have investigated the role of each component of the dynamic search framework introduced in Chapter 3 and answered the first group of research questions presented in Chapter 4. In Section 5.1, we studied the impact of the initial list of documents on the dynamic search system. In particular, we found that precision of the first batch of documents presented to the user impacts the overall system performance as well the recall of later batches. In Section 5.2, we showed that an inaccurate or incomplete aspect modeling impacts the entire search system. In Section 5.3, we showed that the system effectiveness is resilient to noisy aspect coverage estimates for dynamic reranking to a large extent, especially for queries with fewer aspects. Finally, in Section 5.4, we found that stopping late tends to incur more effort than gain. Overall, the results in this chapter demonstrated that dynamic reranking is the single most impactful component for the effectiveness of a dynamic search system. Accordingly, in

the chapter, we discuss preliminary results on practical instantiations of this component, including a breakdown analysis of encompassing different query types, amount of relevant documents, and amount of aspects.

# Chapter 6

# Practical Instantiations

A dynamic search system, as discussed in Chapter 3, needs to interact with the user in order to continually identify aspects of interest that should be covered by the subsequently returned documents. In particular, at time $t$, an aspect modeling component receives the feedback set $\mathcal{F}_t$ from the user and produce an aspect model $\mathcal{A}_t$. Then, a dynamic reranker interprets the aspect model and estimates the aspect coverage matrix that is used to produce a new list of documents $\mathcal{D}_{t+1}$.

In the previous chapter, we showed that our working instantiation of a dynamic search system via interactive search result diversification can generate at each time $t$ a list of documents $\mathcal{D}_t$ that covers a set of aspects that are from the interest of the user. Specifically, in Section 5.3, we showed that the dynamic search system can outperform an ad-hoc vanilla baseline with an aspect coverage estimates above a noised threshold by using the structured user's feedback. Because of that, in this chapter we investigate different instantiations of the dynamic reranker component.

In the remainder of this chapter, we present a preliminary investigation of how to interpret the aspect model $\mathcal{A}_t$ and estimate the aspect coverage matrix. In particular, our experiments aim to answer the following research questions related to *Q2*, "Can a practical instantiation of the framework through interactive diversification improve over a vanilla ad-hoc search baseline?":

*Q2.1* – Can we use external evidence to improve aspect coverage estimates?

*Q2.2* – What queries are improved the most and the least?

In the remainder of this chapter: Section 6.1 describes approaches to estimate an aspect coverage matrix; Section 6.2 presents the experimental setup for this chapter

such as baselines, training and test procedures; and Section 6.3 presents our preliminary results and a breakdown analyses.

## 6.1 Estimating Aspect Coverage

Recall that, for each retrieved document $d$, structured feedbacks $f = \langle p, a, g \rangle$ include a passage $p$ that the user deemed relevant to aspect $a$ at a given relevance level $g$. We aim to estimate the coverage of document $d$ with respect to the aspect $a$, which depends on the type of interactive diversification model, flat or hierarchical. In the remainder of this chapter, Section 6.1.1 presents our first approach to estimate the aspect coverage matrix, and Section 6.1.2, we mine the aspect tree in order to use different external evidence that would help to estimate the coverage of aspects more accurately.

### 6.1.1 Document Passage Relevance

We proposed to estimate the aspect coverage using the relevance of the passage $p$ given a document $d$. In particular, for the flat diversification performed by xQuAD and PM2 (Equation (3.3), and Equation (3.9)), because multiple passages can be deemed relevant to the same aspect, they estimate the coverage of document $d$ with respect to aspect $a$ as follows:

$$P(d|q, a) = \max_{p \in a} P(d|p), \tag{6.1}$$

where $P(d|p)$ denotes the coverage of passage $p$ by document $d$. Before the user's feedback is received, this probability is estimated proportionally to the relevance score between the passage and the document (in our case DPH score). Afterward, this probability is estimated proportionally to the relevance level $g$ directly assigned by the user. In contrast, for HxQuAD and HPM2 (Equation (3.7), and Equation (3.11)), we directly leverage the hierarchical relationship between the aspect $a$ and each of its associated passages $p \in a$ as contributed by the user.

### 6.1.2 Selecting Terms using External Evidence

An aspect model $\mathcal{A}^t$ is composed of several aspects and the document passages that are attached to it. A document passage is a piece of text from a document composed of several terms that belong to one or more aspect. A term $t$ in a passage $p$ could be more relevant than others terms. The relevance of a term to a passage $p$ and aspect $a$ may be

estimated using the collection itself or from external evidence. Several works studied the contribution of terms in feedback documents using external evidence [Bendersky et al., 2010, 2011, 2012; Odijk et al., 2015].

Inspired by the aforementioned works, for flat models, we learn to select the most relevant terms from the passages that belong to an aspect $a$, and for hierarchical models, we learn to select the most relevant from the passage itself. In particular, given a set of passages $\mathcal{P}$ for an aspect $a$, we select a pool of terms $\mathcal{T}$ that are the candidate terms to represent the aspect $a$. Next, $\mathcal{T}$ is refined in a new set of terms $\mathcal{T}^*$. Thus, we estimate the coverage of document $d$ with respect to $a$ as the relevance score between the terms in $\mathcal{T}^*$ and the document. In the following, we describe how we selected $\mathcal{T}$ and refined it to obtain $\mathcal{T}^*$.

First, to obtain the terms $\mathcal{T}$, we select the most relevant terms for the aspect $a$ according to relevance evidence feature $f_i$. For each evidence, we select $N$ terms, joining this terms in $\mathcal{T}$, giving a total of $N \times b$ terms, where $b$ is the number of evidence features, and $N$ the number of terms. For this preliminary investigation, we set $N = 100$, as was proposed by Bendersky et al. [2012].

Next, we refine the $\mathcal{T}$ to obtain $\mathcal{T}^*$ by selecting the $N^*$ most relevant terms according to a linear combination of the evidence features as follows:

$$f(t, q, a) = \sum_{i \in \mathcal{E}} w_i \times f_i, \tag{6.2}$$

where $w_i$ is the parameter weight for feature evidence $i$, and $f_i$ is the value of the normalized feature $f_i$. Similarly to the approach described in 6.1.2, after receiving the feedback in a document, we use the relevance level $g$ directly assigned by the user. Also, for hierarchical models, we directly leverage the hierarchical relationship between the aspect $a$ and each of its attained passages $p \in a$ as provided by the user. In the remainder of this section, we describe the evidence features $\mathcal{E}$ and the weight $w_i$ optimization procedure used in this dissertation.

#### 6.1.2.1 Relevance Evidence Features

In this dissertation, we make use of eight relevance evidence features separated into five groups with some of them inspired by Bendersky et al. [2011]. The first group uses the internal collection features. We extract statistics about a term in the collection and in the feedback passages for an aspect. The remainder four groups use external collections to compute their evidence features. The second group uses the entropy

of a term $t$ in the Google n-grams dataset[1], which gives the evidence of how much information a term convey. The third group uses the frequency of a term in Wikipedia titles[2]. The fourth group uses the frequency of the term in a search log[3]. The fifth group uses the frequency of term as an entity in the passage set. [4] In Table 6.1, we summarize the nine relevance evidence features.

Table 6.1: Evidence features $f_i$.

| Feature $f_i$ | Description |
|---|---|
| $IDF(t)$ | Inverse Document Frequency of $t$ in the collection |
| $PF(t)$ | Frequency of $t$ in the passages set $\mathcal{P}$ |
| $IDFW(t)$ | Inverse Document Frequency of $t$ in the passages set $\mathcal{P}$ weighted by the graded $g$ relevance of passage $p$ |
| $PFW(t)$ | Frequency of $t$ in the passage set $\mathcal{P}$ weighted by the graded $g$ relevance of passage $p$ |
| $GE(t)$ | Entropy of $t$ in Google n-grams |
| $WF(t)$ | Frequency of $t$ in Wikipedia titles |
| $QF(t)$ | Frequency of $t$ in search logs |
| $EF(t)$ | Frequency of $t$ as an entity in the passage set $\mathcal{P}$ |

#### 6.1.2.2 Evidence Weight Optimization

To estimate the weight parameters $w_i$ associated with the evidence features in the ranking function in Equation 6.2, we apply techniques on the learning to rank literature (see Liu [Liu, 2009] for a survey). In this dissertation, we use the coordinate ascent (CA) as an optimization algorithm, which was proposed by Metzler and Bruce Croft [2007]. The CA algorithm iteratively optimizes a target effectiveness metric (in our case, the metric ACT@10) by making a series of one-dimensional line searches. This algorithm cycles through each of the parameters $w_i$, holding all other parameters fixed while optimizing it. For each parameter, a learning step $s$ is used to speed up the process (we set $s = 0.1$). This process is performed iteratively over all parameters until the increase in the target metric is below a certain threshold (we set as $10^4$). During the work of this dissertation, we tried online learning to rank approaches such as DBGB [Yue and Joachims, 2009], but we report only the results for the CA algorithm because of its simplicity, efficiency, and effectiveness.

## 6.2 Experimental Setup

In this section, we describe the baselines used in our investigations and training procedure carried out to learn the hyperparameters and weight optimization procedure.

[1]Available in http://storage.googleapis.com/books/ngrams/books/datasetsv2.html
[2]Available in https://dumps.wikimedia.org/
[3]MSN search query log excerpt (the RFP 2006 dataset).
[4]DBPedia with standard parameters, available in bit.ly/DBpedia-entity

### 6.2.1 Baselines

To answer our research questions of this chapter, we compare our system with three baseline approaches. First, the *vanilla ad-hoc search baseline*, which we use DPH as it showed to be the most effective between the ad-hoc search system.

Besides DPH, we consider as *diversification baseline* the Maximum Marginal Relevance (MMR, [Carbonell and Goldstein, 1998]). MMR is an implicit search result diversification approach, that instantiates the scoring function in Equation (3.1) by estimating the similarity between $d \in \mathcal{R} \setminus \mathcal{D}^t$ and its most dissimilar document $d_j \in \mathcal{D}^t$. Precisely, we have:

$$score_{MMR}(q, d, \mathcal{A}, \mathcal{D}) = \lambda rel_1(q, d) + (1 - \lambda) \max_{d_j \in \mathcal{D}} rel_2(d, d_j), \tag{6.3}$$

where $rel_1(q, d)$ and $rel_2(d, d_j)$ estimate the relevance of $d$ to the query $q$ and its similarity to the documents already in $\mathcal{D}$, respectively. A calibration between relevance $rel_1$ and redundancy (i.e, $\max rel_2$, the opposite of novelty) is achieved through an appropriate setting of $\lambda$, as will be described in Section 6.2.2. In our experiments, $rel_1$ by DPH retrieval model. Following Carbonell and Goldstein [1998], we compute $rel_2$ as the cosine between the implicit representations of $d$ and $d_j$ (vector space model with TFIDF weight scheme).

Finally, we consider as *relevance feedback baseline* the Relevance Model RM3 proposed by Lavrenko and Croft [2001]. In this model, given the passages in the feedback set $\mathcal{F}^t$, it reformulates the original query $q$ into $q_m$ with expansion *terms* and their *weights*. In particular, for each term $w \in W$ in the feedback set $\mathcal{F}^t$, we have:

$$P(w|\theta_{\mathcal{F}}) = \lambda\, P(w|\theta_q) + \sum_{d \in \mathcal{F}} \frac{P(q|d)}{\mathcal{Z}}\, P(w|d), \tag{6.4}$$

where $P(q|d)$ is the score of document $d$ in the initial ranking $\mathcal{R}$ for query $q$, $\mathcal{Z} = d \in \mathcal{F}\, P(q|d)$ serves to normalize the retrieval scores. $P(w|d)$ is estimated using the maximum likelihood estimate (MLE) as $\frac{c(w,d)}{|d|}$, and the query likelihood model $P(w|\theta_q)$ is estimated as $\frac{c(w,q)}{|d|}$. A calibration between initial term relevance and feedback term relevance is achieved through an appropriate setting of $\lambda$, as will be described in Section 6.2.2. Finally, $q_m$ is build by selecting the top $n$ terms according to the query language model $P(w|\theta_{\mathcal{F}})$ and the new scores for the documents in $\mathcal{R}$.

### 6.2.2 Training and Test Procedure

As discussed in Chapter 3, the interactive search result diversification approaches, the diversification baseline, and relevance feedback baseline are parameterized by $\lambda$. In addition, the relevance feedback baseline has $n$, the number of terms $n$ to expand, as hyperparameter. In our experiments, in order to train $\lambda$ and $n$, we perform a 5-fold cross-validation over the queries for each domain, optimizing for ACT@10. We vary $\lambda$ with step 0.1 within the range of [0.1,0.9], and vary $n$ with step 10 within the range [10,100]. To find these parameters, we use only the validation folder as we use the training folder to learn $w_i$ and the same validation folder to find the hyperparameters. To check for statistically significant differences among the baselines, we use a two-tailed paired t-test and write ▲ (▼) and △ (▽) denote significant increases (decreases) at the 0.01 and 0.05 $p$-values, respectively. A further symbol ∘ is used to denote no significant difference.

## 6.3 Experimental Results

In this chapter, we hypothesize that our system may benefit from a coverage estimate learned using external evidence. To verify this hypothesis, we address the research question *Q2.1*, by assessing our system using document passage relevance versus using a term expansion with external evidence as described Section 6.1.2. Table 6.2, 6.3, and 6.4 summarize our results for the TREC 2015-2016 Dynamic Domain track queries for ACT $\alpha$-nDCG, and ERR-IA, respectively. In each row describing baseline results (the first three rows of the table), the first of the symbols introduced in Section 6.2.2 denotes a statistically significant difference with respect to DPH, a second symbol denotes potential differences with respect to DPH+MMR, and a third symbol denotes potential difference against DPH+RM3.

From Table 6.2, 6.3, and 6.4, we first observe that the interactive diversification approaches show to outperform the diversification and relevance feedback baselines in the some cases. However, diversification and relevance feedback baselines do not outperform a vanilla ad-hoc search system. However, a challenge is to outperform such vanilla system, which does not use the user's feedback to improve their search results. Nonetheless, using the document passage relevance to estimating the aspect coverage matrix in interactive diversification approaches show some significant improvements over the vanilla baseline. In particular, for the test collections of 2015, we observe that xQuAD improves up to 1.8% in $\alpha$-nDCG@10 and 1% in ERR-IA@10. Moreover, for the test collection of 2016, xQuAD using document passage relevance, we observe a

Table 6.2: Comparison of DPH, DPH+MMR, DPH+RMR, and interactive search result diversification with document passage relevance and selecting relevant terms using external evidence in terms of ACT.

| | 2015 | | | 2016 | | |
|---|---|---|---|---|---|---|
| | @1 | @2 | @10 | @1 | @2 | @10 |
| DPH | $0.2283^{\circ\blacktriangle\blacktriangle}$ | $0.1989^{\circ\blacktriangle\blacktriangle}$ | $0.1039^{\circ\blacktriangle\blacktriangle}$ | $0.1789^{\circ\circ\circ}$ | $0.1600^{\circ\circ\circ}$ | $0.0868^{\circ\circ\circ}$ |
| +MMR | $0.2091^{\blacktriangledown\circ\circ}$ | $0.1814^{\blacktriangledown\circ\circ}$ | $0.0968^{\blacktriangledown\circ\circ}$ | $0.1724^{\circ\circ\circ}$ | $0.1531^{\circ\circ\circ}$ | $0.0841^{\circ\circ\circ}$ |
| +RM3 | $0.2283^{\circ\blacktriangle\blacktriangle}$ | $0.1978^{\circ\blacktriangle\blacktriangle}$ | $0.1023^{\blacktriangledown\blacktriangle\blacktriangle}$ | $0.1789^{\circ\circ\circ}$ | $0.1613^{\circ\circ\circ}$ | $0.0862^{\circ\circ\circ}$ |
| Document Passage Relevance | | | | | | |
| +xQuAD | $0.2283^{\circ\blacktriangle\blacktriangle}$ | $0.1996^{\circ\blacktriangle\blacktriangle}$ | $0.1056^{\circ\blacktriangle\blacktriangle}$ | $0.1789^{\circ\circ\circ}$ | $0.1618^{\circ\circ\circ}$ | $0.0901^{\triangle\blacktriangle\blacktriangle}$ |
| +HxQuAD | $0.2283^{\circ\blacktriangle\blacktriangle}$ | $0.2001^{\circ\blacktriangle\blacktriangle}$ | $0.1059^{\circ\blacktriangle\blacktriangle}$ | $0.1789^{\circ\circ\circ}$ | $0.1616^{\circ\circ\circ}$ | $0.0895^{\circ\triangle\triangle}$ |
| +PM2 | $0.2283^{\circ\blacktriangle\blacktriangle}$ | $0.1981^{\circ\blacktriangle\blacktriangle}$ | $0.1033^{\circ\triangle\triangle}$ | $0.1789^{\circ\circ\circ}$ | $0.1618^{\circ\circ\circ}$ | $0.0888^{\circ\circ\circ}$ |
| +HPM2 | $0.2283^{\circ\blacktriangle\blacktriangle}$ | $0.1982^{\circ\blacktriangle\blacktriangle}$ | $0.1025^{\circ\triangle\triangle}$ | $0.1789^{\circ\circ\circ}$ | $0.1614^{\circ\circ\circ}$ | $0.0879^{\circ\circ\circ}$ |
| Selecting Relevant Terms | | | | | | |
| +xQuAD | $0.2283^{\circ\blacktriangle\blacktriangle}$ | $0.2009^{\circ\blacktriangle\blacktriangle}$ | $0.1057^{\circ\blacktriangle\blacktriangle}$ | $0.1789^{\circ\circ\circ}$ | $0.1604^{\circ\circ\circ}$ | $0.0889^{\circ\triangle\triangle}$ |
| +HxQuAD | $0.2283^{\circ\blacktriangle\blacktriangle}$ | $0.2007^{\circ\blacktriangle\blacktriangle}$ | $0.1068^{\triangle\blacktriangle\blacktriangle}$ | $0.1789^{\circ\circ\circ}$ | $0.1617^{\circ\circ\circ}$ | $0.0899^{\circ\triangle\triangle}$ |
| +PM2 | $0.2283^{\circ\blacktriangle\blacktriangle}$ | $0.1977^{\circ\blacktriangle\blacktriangle}$ | $0.1016^{\circ\triangle\triangle}$ | $0.1789^{\circ\circ\circ}$ | $0.1621^{\circ\circ\circ}$ | $0.0871^{\circ\circ\circ}$ |
| +HPM2 | $0.2283^{\circ\blacktriangle\blacktriangle}$ | $0.1978^{\circ\blacktriangle\blacktriangle}$ | $0.1026^{\circ\triangle\triangle}$ | $0.1789^{\circ\circ\circ}$ | $0.1599^{\circ\circ\circ}$ | $0.0870^{\circ\circ\circ}$ |

Table 6.3: Comparison of DPH, DPH+MMR, DPH+RMR, and interactive search result diversification with document passage relevance and selecting relevant terms using external evidence in terms of $\alpha$-nDCG.

| | 2015 | | | 2016 | | |
|---|---|---|---|---|---|---|
| | @1 | @2 | @10 | @1 | @2 | @10 |
| DPH | $0.3585^{\circ\blacktriangle\blacktriangle}$ | $0.3930^{\circ\blacktriangle\blacktriangle}$ | $0.4677^{\circ\blacktriangle\blacktriangle}$ | $0.3483^{\circ\circ\circ}$ | $0.3734^{\circ\circ\circ}$ | $0.4561^{\circ\circ\circ}$ |
| +MMR | $0.3288^{\blacktriangledown\circ\circ}$ | $0.3624^{\blacktriangledown\circ\circ}$ | $0.4388^{\blacktriangledown\circ\circ}$ | $0.3452^{\circ\circ\circ}$ | $0.3671^{\circ\circ\circ}$ | $0.4507^{\circ\circ\circ}$ |
| +RM3 | $0.3585^{\circ\blacktriangle\blacktriangle}$ | $0.3866^{\triangledown\triangle\triangle}$ | $0.4619^{\triangledown\blacktriangle\blacktriangle}$ | $0.3483^{\circ\circ\circ}$ | $0.3799^{\circ\circ\circ}$ | $0.4513^{\circ\circ\circ}$ |
| Document Passage Relevance | | | | | | |
| +xQuAD | $0.3585^{\circ\blacktriangle\blacktriangle}$ | $0.3966^{\circ\blacktriangle\blacktriangle}$ | $0.4760^{\triangle\blacktriangle\blacktriangle}$ | $0.3483^{\circ\circ\circ}$ | $0.3844^{\circ\circ\circ}$ | $0.4752^{\blacktriangle\blacktriangle\blacktriangle}$ |
| +HxQuAD | $0.3585^{\circ\blacktriangle\blacktriangle}$ | $0.3975^{\circ\blacktriangle\blacktriangle}$ | $0.4760^{\circ\blacktriangle\blacktriangle}$ | $0.3483^{\circ\circ\circ}$ | $0.3835^{\circ\circ\circ}$ | $0.4706^{\triangle\triangle\triangle}$ |
| +PM2 | $0.3585^{\circ\blacktriangle\blacktriangle}$ | $0.3899^{\circ\triangle\triangle}$ | $0.4656^{\circ\blacktriangle\blacktriangle}$ | $0.3483^{\circ\circ\circ}$ | $0.3806^{\circ\circ\circ}$ | $0.4634^{\circ\circ\circ}$ |
| +HPM2 | $0.3585^{\circ\blacktriangle\blacktriangle}$ | $0.3942^{\circ\blacktriangle\blacktriangle}$ | $0.4610^{\circ\triangle\triangle}$ | $0.3483^{\circ\circ\circ}$ | $0.3770^{\circ\circ\circ}$ | $0.4516^{\circ\circ\circ}$ |
| Selecting Relevant Terms | | | | | | |
| +xQuAD | $0.3585^{\circ\blacktriangle\blacktriangle}$ | $0.4014^{\triangle\blacktriangle\blacktriangle}$ | $0.4759^{\triangle\blacktriangle\blacktriangle}$ | $0.3483^{\circ\circ\circ}$ | $0.3779^{\circ\circ\circ}$ | $0.4629^{\circ\circ\circ}$ |
| +HxQuAD | $0.3585^{\circ\blacktriangle\blacktriangle}$ | $0.3998^{\circ\blacktriangle\blacktriangle}$ | $0.4811^{\blacktriangle\blacktriangle\blacktriangle}$ | $0.3483^{\circ\circ\circ}$ | $0.3850^{\circ\circ\circ}$ | $0.4723^{\circ\circ\circ}$ |
| +PM2 | $0.3585^{\circ\blacktriangle\blacktriangle}$ | $0.3920^{\circ\blacktriangle\blacktriangle}$ | $0.4594^{\circ\triangle\triangle}$ | $0.3483^{\circ\circ\circ}$ | $0.3796^{\circ\circ\circ}$ | $0.4521^{\circ\circ\circ}$ |
| +HPM2 | $0.3585^{\circ\blacktriangle\blacktriangle}$ | $0.3903^{\circ\triangle\triangle}$ | $0.4630^{\circ\triangle\triangle}$ | $0.3483^{\circ\circ\circ}$ | $0.3748^{\circ\circ\circ}$ | $0.4471^{\circ\circ\circ}$ |

Table 6.4: Comparison of DPH, DPH+MMR, DPH+RMR, and interactive search result diversification with document passage relevance and selecting relevant terms using external evidence in terms of ERR-IA.

| | 2015 | | | 2016 | | |
|---|---|---|---|---|---|---|
| | @1 | @2 | @10 | @1 | @2 | @10 |
| DPH | $0.2786^{\circ\blacktriangle\blacktriangle}$ | $0.2953^{\circ\blacktriangle\blacktriangle}$ | $0.3105^{\circ\blacktriangle\blacktriangle}$ | $0.2538^{\circ\circ\circ}$ | $0.2673^{\circ\circ\circ}$ | $0.2854^{\circ\circ\circ}$ |
| +MMR | $0.2601^{\blacktriangledown\circ\circ}$ | $0.2756^{\blacktriangledown\circ\circ}$ | $0.2910^{\blacktriangledown\circ\circ}$ | $0.2504^{\circ\circ\circ}$ | $0.2625^{\circ\circ\circ}$ | $0.2799^{\circ\circ\circ}$ |
| +RM3 | $0.2786^{\circ\blacktriangle\blacktriangle}$ | $0.2931^{\triangledown\blacktriangle\blacktriangle}$ | $0.3082^{\triangledown\blacktriangle\blacktriangle}$ | $0.2538^{\circ\circ\circ}$ | $0.2700^{\circ\circ\circ}$ | $0.2850^{\circ\circ\circ}$ |
| Document Passage Relevance | | | | | | |
| +xQuAD | $0.2786^{\circ\blacktriangle\blacktriangle}$ | $0.2971^{\circ\blacktriangle\blacktriangle}$ | $0.3135^{\triangle\blacktriangle\blacktriangle}$ | $0.2538^{\circ\circ\circ}$ | $0.2715^{\circ\circ\circ}$ | $0.2914^{\triangle\circ\circ}$ |
| +HxQuAD | $0.2786^{\circ\blacktriangle\blacktriangle}$ | $0.2975^{\circ\blacktriangle\blacktriangle}$ | $0.3137^{\circ\blacktriangle\blacktriangle}$ | $0.2538^{\circ\circ\circ}$ | $0.2710^{\circ\circ\circ}$ | $0.2898^{\circ\circ\circ}$ |
| +PM2 | $0.2786^{\circ\blacktriangle\blacktriangle}$ | $0.2946^{\circ\blacktriangle\blacktriangle}$ | $0.3105^{\circ\blacktriangle\blacktriangle}$ | $0.2538^{\circ\circ\circ}$ | $0.2707^{\circ\circ\circ}$ | $0.2884^{\circ\circ\circ}$ |
| +HPM2 | $0.2786^{\circ\blacktriangle\blacktriangle}$ | $0.2957^{\circ\blacktriangle\blacktriangle}$ | $0.3099^{\circ\blacktriangle\blacktriangle}$ | $0.2538^{\circ\circ\circ}$ | $0.2691^{\circ\circ\circ}$ | $0.2856^{\circ\circ\circ}$ |
| Selecting Relevant Terms | | | | | | |
| +xQuAD | $0.2786^{\circ\blacktriangle\blacktriangle}$ | $0.2990^{\triangle\blacktriangle\blacktriangle}$ | $0.3143^{\blacktriangle\blacktriangle\blacktriangle}$ | $0.2538^{\circ\circ\circ}$ | $0.2692^{\circ\circ\circ}$ | $0.2885^{\circ\circ\circ}$ |
| +HxQuAD | $0.2786^{\circ\blacktriangle\blacktriangle}$ | $0.2985^{\circ\blacktriangle\blacktriangle}$ | $0.3151^{\blacktriangle\blacktriangle\blacktriangle}$ | $0.2538^{\circ\circ\circ}$ | $0.2716^{\circ\circ\circ}$ | $0.2908^{\circ\circ\circ}$ |
| +PM2 | $0.2786^{\circ\blacktriangle\blacktriangle}$ | $0.2952^{\circ\blacktriangle\blacktriangle}$ | $0.3092^{\circ\blacktriangle\blacktriangle}$ | $0.2538^{\circ\circ\circ}$ | $0.2703^{\circ\circ\circ}$ | $0.2858^{\circ\circ\circ}$ |
| +HPM2 | $0.2786^{\circ\blacktriangle\blacktriangle}$ | $0.2945^{\circ\blacktriangle\blacktriangle}$ | $0.3093^{\circ\blacktriangle\blacktriangle}$ | $0.2538^{\circ\circ\circ}$ | $0.2677^{\circ\circ\circ}$ | $0.2838^{\circ\circ\circ}$ |

significant improvement of up to 3.6% in ACT@10, 4.0% in $\alpha$-nDCG@10, and 2.0% in ERR-IA@10. Furthermore, for the HxQuAD we observe improvements up to 3% in $\alpha$-nDCG@10.

Recalling question *Q2.1*, from Table 6.2, 6.3, and 6.4, we observe an overall improvement using the selecting relevant terms for the 2015 test collection. In particular, in comparison with a vanilla ad-hoc baseline, we observe that xQuAD improves up to 2% in $\alpha$-nDCG@2, 1.7% in $\alpha$-nDCG@10, 1.3% in both ERR-IA@2, and ERR-IA@10. Also, we observe that HxQUAD improves up to 2.8% in ACT@10, 3.0% in $\alpha$-nDCG@10, and 2.5% in ERR-IA@10. It is worth mention that the improvements using an selecting relevant terms are higher than using document passage relevance, which mostly answers $Q2.1$. In contrast, for the 2016 test collection, we observe that an selecting relevant terms does not help to improve over a vanilla ad-hoc search baseline.

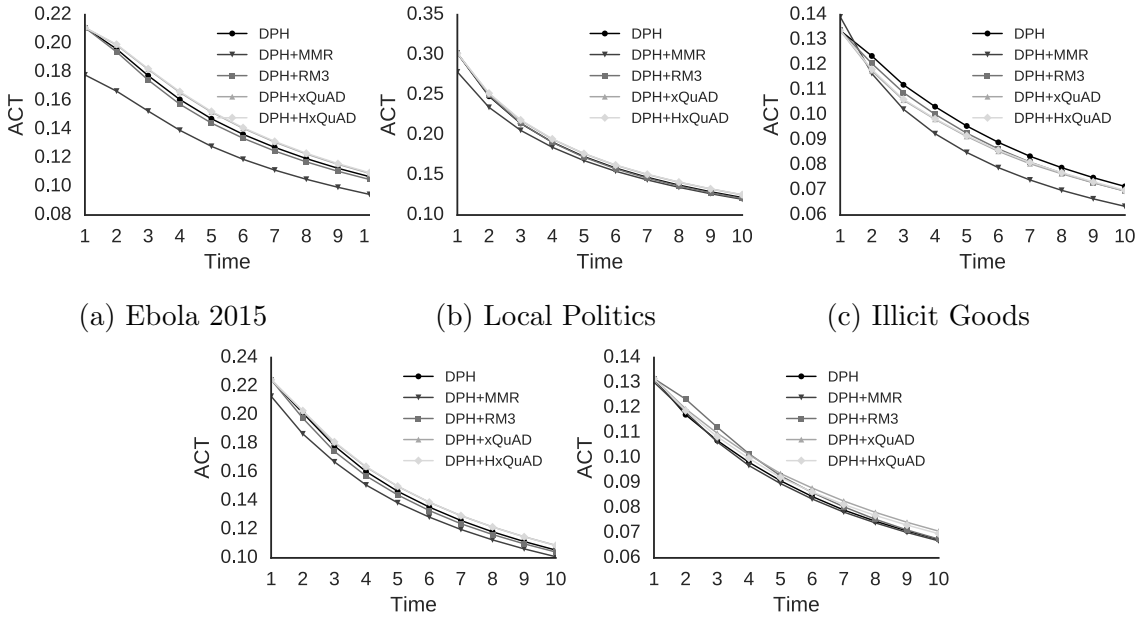
### 6.3.1 Breakdown Analyses

The results present in Table 6.2, 6.3, and 6.4, show little improvements over a vanilla ad-hoc baseline on the entire set of queries for the TREC 2015-2016 Dynamic Domain track test collections. In the remainder of this section, we address *Q2.2*, by assessing the breakdown effectiveness analyses of the interactive search result diversification in contrast to the various baselines described in Section 6.2.1. To further shed light on the reasons behind where we can improve our results, we analyze the improvements

brought by xQuAD, and HxQuAD using document passage relevance, as proposed by Moraes et al. [2016], for queries with different domains, type, and aspect set size. Last, we show the results individually by query.

#### 6.3.1.1 Analysis by Domain

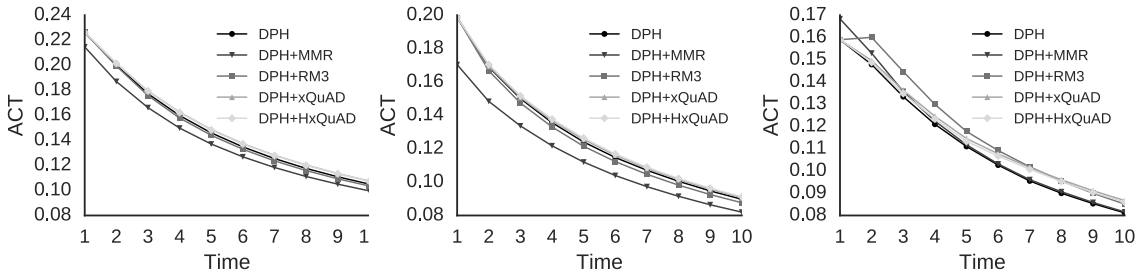
The TREC Dynamic Domain track encompasses the challenge of searching in specialized domains. Figure 6.1 (a-e) show ACT for each time $t$, whereas Figure 6.1 (a-c) are the domains in the 2015 collection and, Figure 6.1 (d-e) are the results for the 2016 test collection. From these figures, we can first observe that Ebola 2015 domain has the highest improvement over the baselines using xQuAD, and HxQuAD. Moreover, the worst result is for the Illicit Goods domain, which any of the approaches can outperform the vanilla ad-hoc search baseline. A possible explanation is that, this domain has different characteristics such as the assessing process (less relevant documents per query), and the nature of the documents of this collection (underground forums on the Web). Surprisingly, we can observe improvements of a relevance feedback baseline, and diversification baselines for the Polar domain over DPH, with xQuAD and HxQuAD slightly improving over it at the end of the interactions.




(a) Ebola 2015 (b) Local Politics (c) Illicit Goods

(d) Ebola 2016 (e) Polar

Figure 6.1: Effectiveness breakdown by domain.




(a) Short (b) Verbose (c) Question Answering

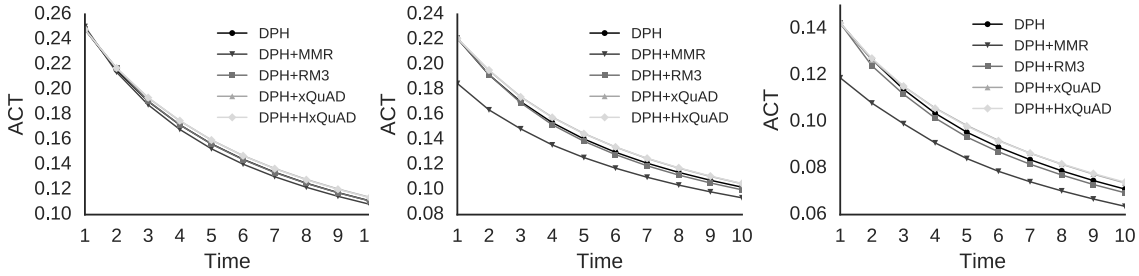
Figure 6.2: Effectiveness breakdown by query type.

#### 6.3.1.2 Analysis by Query Length

A query is the translation of the user's information needs, however, some information needs inquiry longer queries and search systems may perform different depending on the query length. Figure 6.2 (a-c) show the system effectiveness in terms of ACT for each time $t$, for short queries (queries with less than five terms, a total of 106 queries), verbose queries (queries with more than four terms, a total of 53 queries), and question answering queries (queries with question marks and disclosure question marks, a total of 12 queries ). From these figures, we can first observe that the interactive diversification approaches slightly improves over short queries than verbose queries. We can also observe that for question answering queries for the interactive diversification models and the relevance feedback baseline show the highest improvement, which is explained because question answering queries come mostly from the Polar domain, and the nature of documents in this domain (scientific websites from polar sciences) makes possible the improvements over a vanilla ad-hoc search baseline.

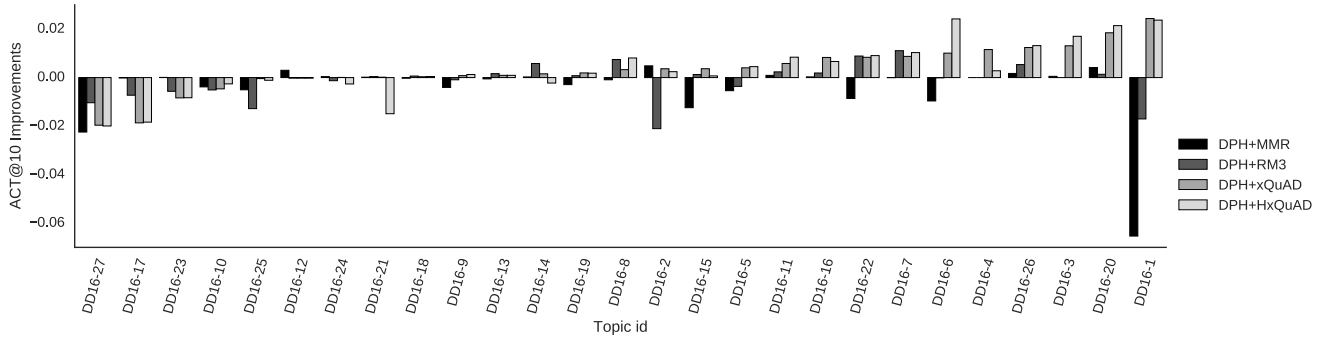
#### 6.3.1.3 Analysis by Aspect Size

Continuing our detailed analysis, in Figure 6.3 breaks down the system effectiveness results in terms of ACT for each time $t$ from Table 6.2, 6.3, and 6.4 according to the amount of aspects a query contains. In particular, we consider three group of queries: low queries, with 1 to 3 aspects (58 queries); medium queries, with 4 to 7 aspects (81 queries); and high queries, with 8 to 17 aspects (32 queries). From Figure 6.3, we observe relatively higher performances of xQuAD, and HxQuAD on high queries compared to those of other sizes of aspects. We also observe that, for low queries, the baselines are closer to xQuAD and HxQuAD.

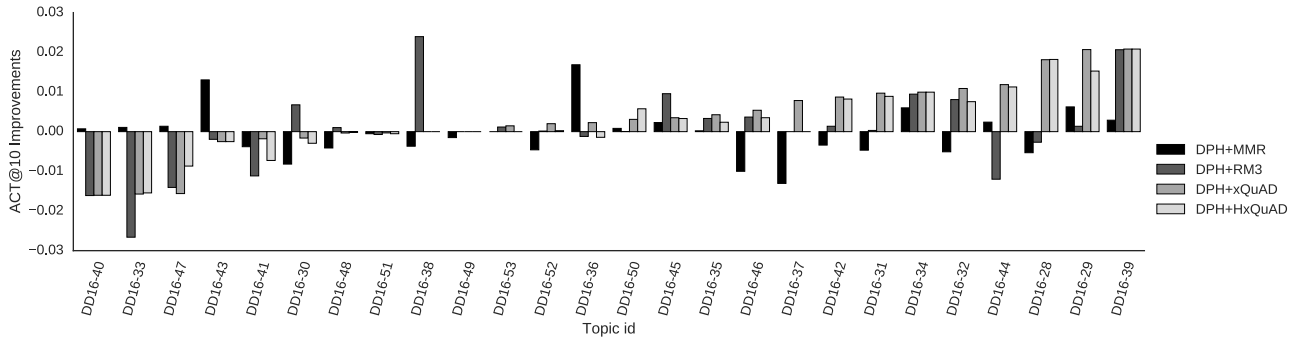



(a) Low (b) Medium (c) High

Figure 6.3: Effectiveness breakdown by aspect size.




(a) Ebola 2016




(b) Polar

Figure 6.4: Differences in ACT@10 between DPH vanilla ad-hoc search baseline and other baselines, and interactive diversification approaches.

#### 6.3.1.4 Analysis by Improvements over Ad-hoc Search Systems

Finally, we perform a detailed analysis of individual queries. In Figure 6.4, we report ACT@10 improvements over the DPH vanilla ad-hoc search baseline for the TREC 2016 Dynamic Domain queries (see Appendix B for TREC 2015 Dynamic Domain queries). From these figures, we can first observe that, in terms of a number of queries, more queries improvements with Ebola domain (19/27) comparing to the Polar domain (16/26). For the queries DD16-20, and DD16-16, we observe that we achieve improvements, which are queries that were alleged by Grace Hui Yang [2016] as the most difficult and easiest topic queries in the TREC 2016 Dynamic Domain track, respectively. This result shows that our interactive diversification can improve over queries that have different level of difficulty.

## 6.4 Summary

In this chapter, we tested the effectiveness of multiple baselines and compared their results against interactive search result diversification approaches, which demonstrated to achieve a promising result while estimating aspects coverage using external evidence. In addition, we explored the results in a breakdown analyses for domains, query types, and query aspects size. Also, we analyze queries improvements individually. We summarize the results found in this chapter as follows:

- Our results using a hierarchical interactive search result diversification showed improvements over flat versions;
- We showed that using external evidence can be promising to improve the system effectiveness for different evaluation metrics;
- From the breakdown analyses we tend to perform better with: Ebola and Polar domains; verbose queries and question answering queries; and queries with larger aspects set.

With this chapter, we conclude the experimental analysis of the dynamic search system framework. In the next chapter, we recap on the contributions of this dissertation, and discuss several directions to continue our work.

# Chapter 7

# Conclusions and Future Work

Information retrieval systems allow users to find facts, to learn, and to help in making decisions through the exploration of the search results using devices such as their desktops or mobile devices. The need for exploration arises in tasks where the user's need is fulfilled through complex search sessions, requiring multiple interactions between the user and search system. During the search process, the search system should dynamically improve its understanding of the user's need via the feedback provided by the user.

In this dissertation, we tackled the dynamic search task as framed by the TREC 2015-2016 Dynamic Domain track. In this task, the system needs (1) provide an initial ranking of candidate results given the user's information need and the domain of interest; (2) decide whether the user's information need has been fulfilled and eventually stop the interactive process; (3) leverage the user's feedback to improve its model of the aspects underlying the users query; (4) produce an improved ranking given the learned aspect model.

We presented a general framework that describes state-of-the-art dynamic search systems that participated in the TREC 2015-2016 Dynamic Domain track campaigns. Through a series of simulations and practical instantiations, we studied the impact of each component of the framework instantiated via interactive search result diversification. In particular, we modified the state-of-the-art search result diversification approaches to interactively discover the various aspects underlying the users initial query.

The following sections summarize the conclusions drawn from our investigation and the main contributions of this work, directions for future works, and our final remarks.

## 7.1 Summary of Contributions

We summarize the main contributions of this dissertation as follows.

- **Simulation Results:** In Chapter 5, we performed a series of simulations to evaluate the impact of different components on the effectiveness of the whole dynamic search system. In particular, in Section 5.1, we studied the impact of the initial list of documents in the system and we found that its precision impacts effectiveness at the first iteration whereas its recall impacts the system effectiveness towards later iterations. In Section 5.2, we showed that an incomplete aspect modeling (i.e., with missing aspects) is less harmful to the effectiveness of the system than an inaccurate aspect modeling (i.e., with missing passages). In Section 5.3, we showed that the system effectiveness is resilient to noisy aspect coverage estimates for dynamic reranking to a large extent, especially for queries with fewer aspects. Finally, in Section 5.4, we found that the additional gain attained by late stopping strategies does not offset the incurred effort.
- **Practical Instantiations:** In Chapter 6, we present a preliminary investigation on practical instantiations of the dynamic reranker component via interactive search result diversification. In particular, we tested the effectiveness of multiple baselines and compared their results against interactive search result diversification approaches, showed promising results while estimating aspects coverage using external evidence. In addition, we explored the results in a breakdown analyses over different domains, query types, and amount of query aspects. Moreover, we analyzed improvements for individual queries.

## 7.2 Summary of Conclusions

In this dissertation, we presented a general framework for dynamic search system in specialized domains. To understand the limitations of each component of this framework, we performed a series of simulations in each component. From these simulations, we showed that a typical dynamic search system via interactive search result diversification can improve the effectiveness of vanilla ad-hoc retrieval models. In addition, we provided preliminary results on practical instantiations of the dynamic reranker component that support the following conclusions.

Precisely, 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 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.

## 7.3 Directions for Future Research

Derived from the research that we conducted during this dissertation, we propose the following directions for future research:

- During our experimental analysis, other instantiations of the framework were investigated, however with limited success. Briefly, we have explored the use of query suggestions from commercial search engines in order to find the query aspects a user may be interested in before any user feedback is received. Similarly, we tried to sample query aspects from latent topic models. As for future work, we believe that enhancing the static sets of aspects mined from query logs or topic models by leveraging the users feedback.

- Another direction is to investigate approaches that learn the query aspects from the user feedback alone. We have tried offline and online learning approaches. From our preliminary experiments not included in this dissertation, online learning approaches in which the weights of a model could be learned quickly to improve the accuracy and completeness of the query aspects seems promising.

- An assumption made along this work was that all the feedbacks from the users are equally important. Another approach is to manage feedbacks in a temporal manner, which could downweight older feedback from the user in order to derive the query aspects.

- On the evaluation side, we believe that dynamic search systems should be evaluated with metrics that encompass gain of information and effort of finding the information. Therefore, another line of work is to better understand the gain-effort trade-off in order to devise more meaningful evaluation metrics.

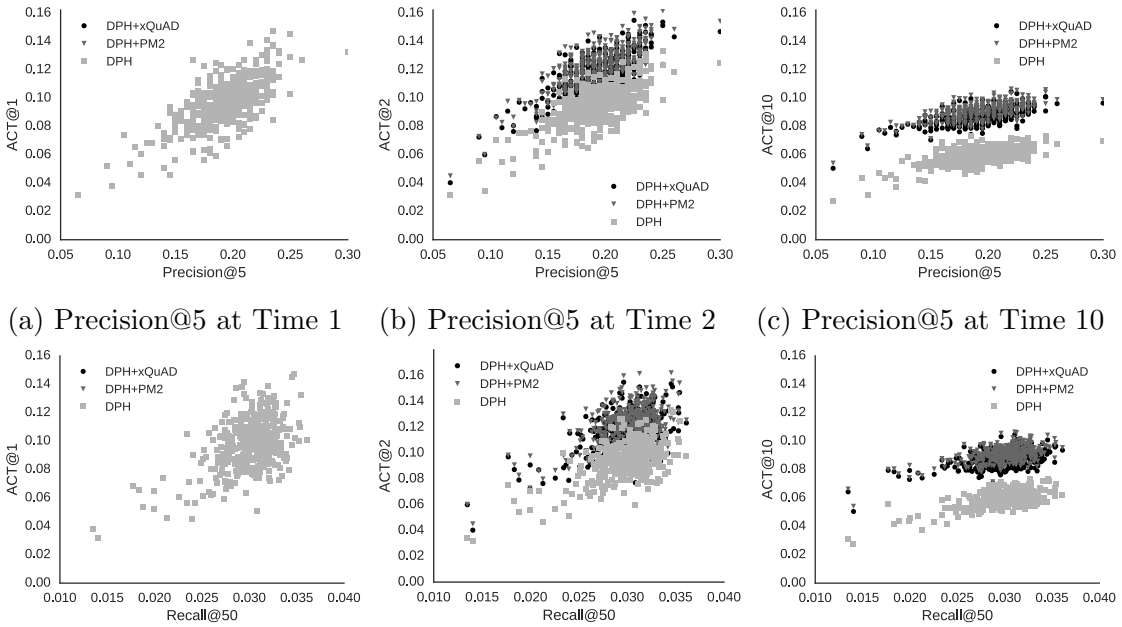
## 7.4 Final Remarks

This dissertation contributed to a better understanding of dynamic search via interactive search result diversification. From a research perspective, this work provided an improved comprehension of the role of different components of a dynamic search system. Furthermore, we participated in the TREC 2016 Dynamic Domain track. Our investigations led to the publication of a paper report in the proceedings of TREC [Moraes et al., 2016]. In addition, our results on interactive search result diversification ranked high across iterations compared with the runs submitted by other participants in this track [Grace Hui Yang, 2016].

# Appendix A

# Simulation Results

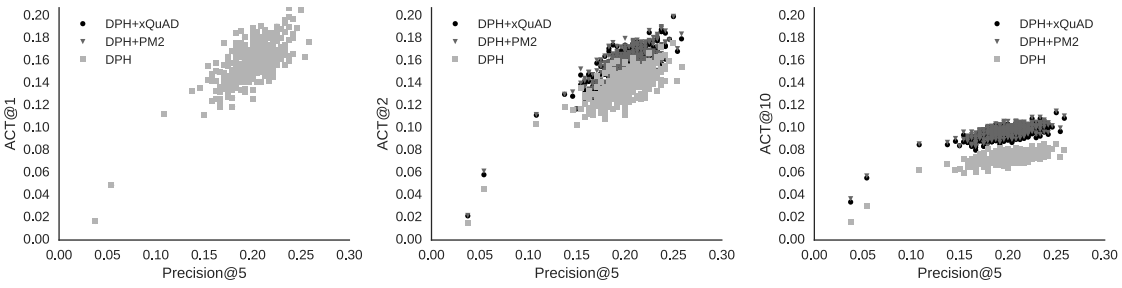
## A.1 Baseline Ranker



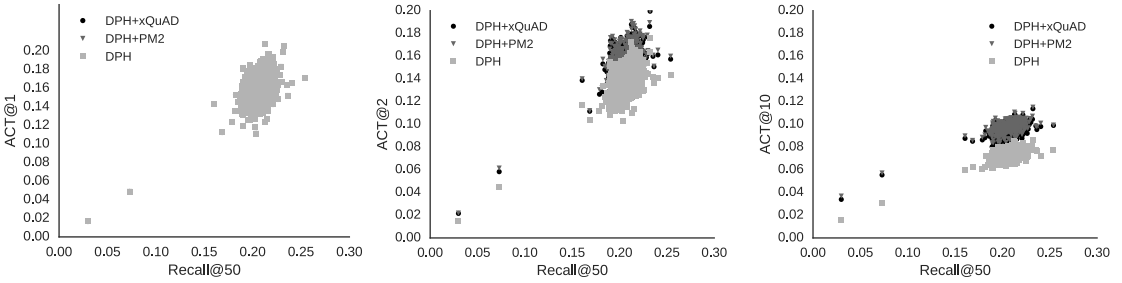

(a) Precision@5 at Time 1 (b) Precision@5 at Time 2 (c) Precision@5 at Time 10

(d) Recall@50 at Time 1 (e) Recall@50 at Time 2 (f) Recall@50 at Time 10

Figure A.1: Ebola 2015 domain - Impact of baseline rankers of various quality levels at times $t \in \{1, 2, 10\}$.

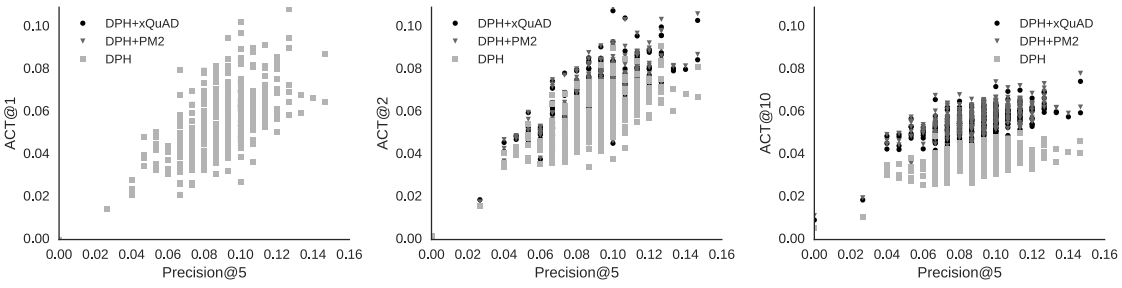



(a) Precision@5 at Time 1 (b) Precision@5 at Time 2 (c) Precision@5 at Time 10

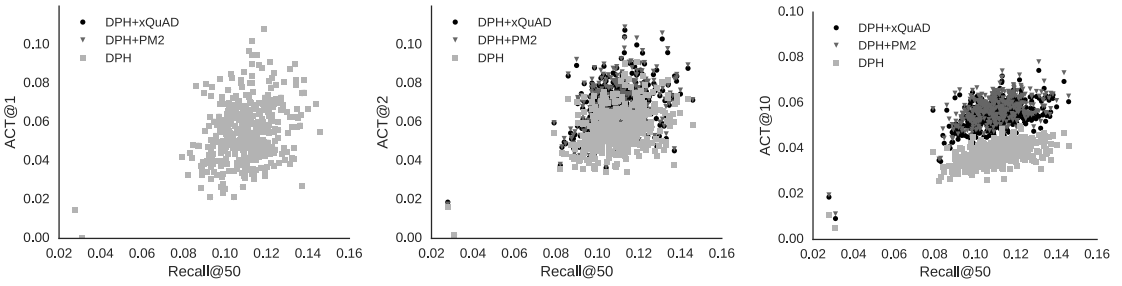



(d) Recall@50 at Time 1 (e) Recall@50 at Time 2 (f) Recall@50 at Time 10

Figure A.2: Local Politics domain - Impact of baseline rankers of various quality levels at times $t \in \{1, 2, 10\}$.

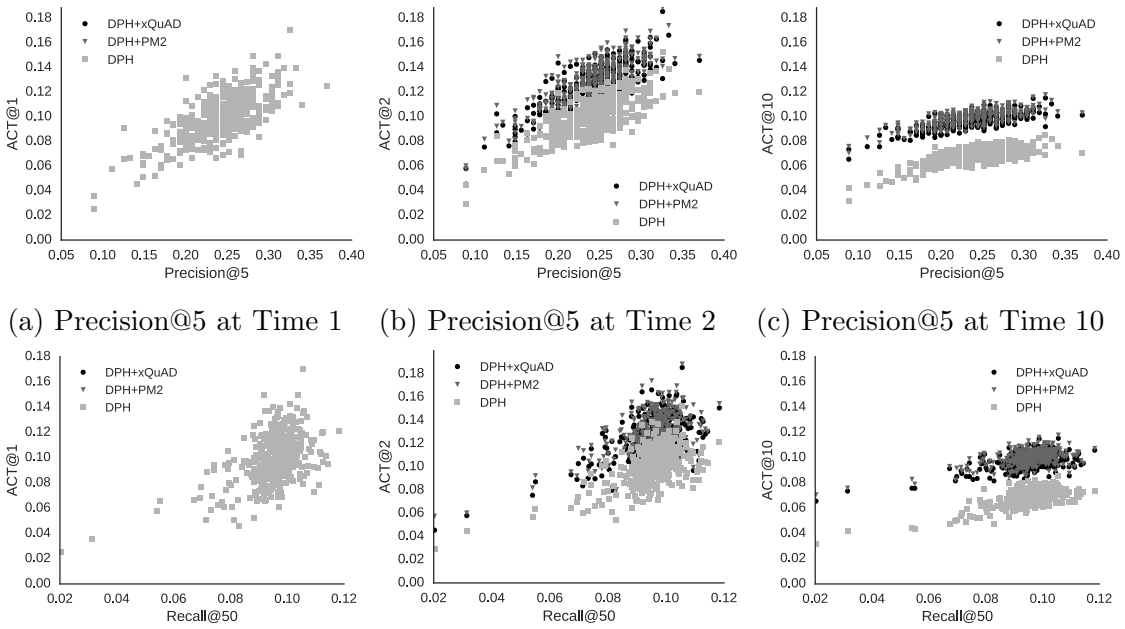



(a) Precision@5 at Time 1 (b) Precision@5 at Time 2 (c) Precision@5 at Time 10




(d) Recall@50 at Time 1 (e) Recall@50 at Time 2 (f) Recall@50 at Time 10

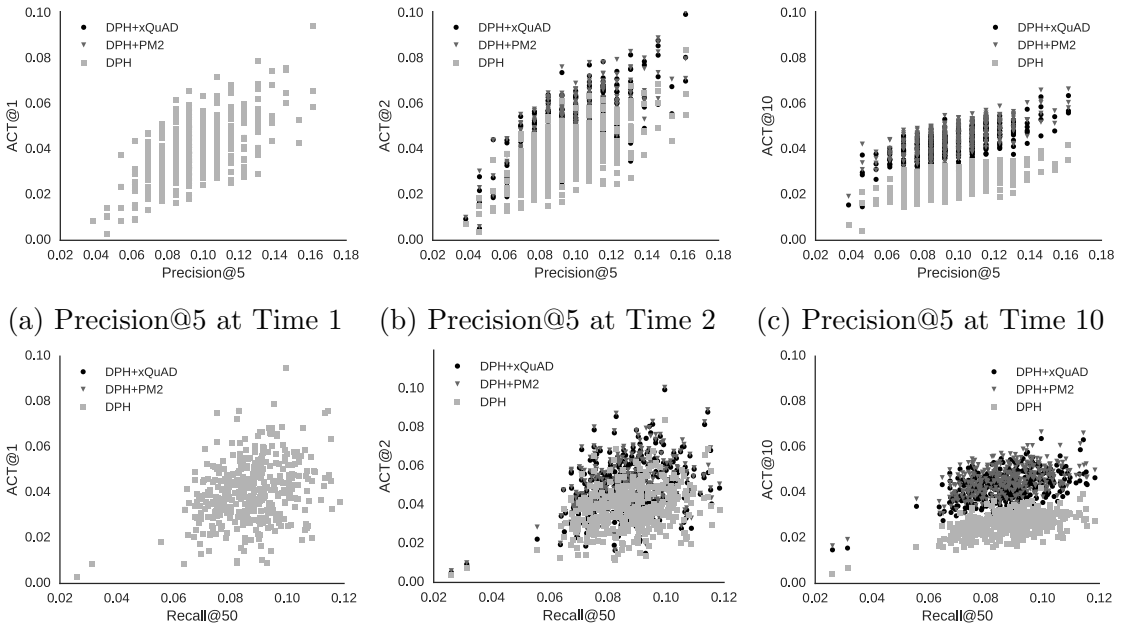
Figure A.3: Illicit Goods domain - Impact of baseline rankers of various quality levels at times $t \in \{1, 2, 10\}$.




(a) Precision@5 at Time 1 (b) Precision@5 at Time 2 (c) Precision@5 at Time 10

(d) Recall@50 at Time 1 (e) Recall@50 at Time 2 (f) Recall@50 at Time 10

Figure A.4: Ebola 2016 domain - Impact of baseline rankers of various quality levels at times $t \in \{1, 2, 10\}$.

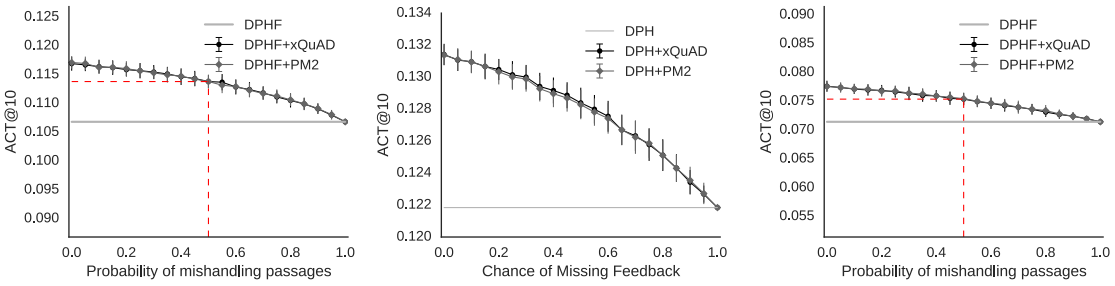



(a) Precision@5 at Time 1 (b) Precision@5 at Time 2 (c) Precision@5 at Time 10

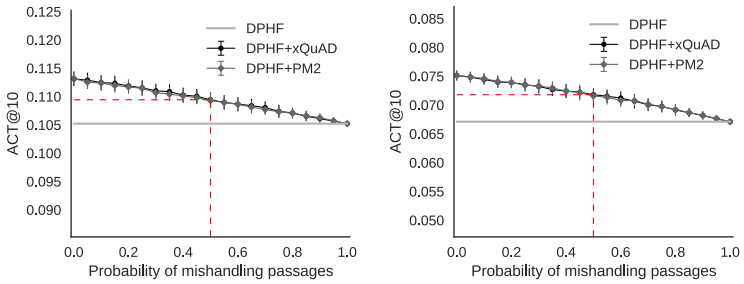
(d) Recall@50 at Time 1 (e) Recall@50 at Time 2 (f) Recall@50 at Time 10

Figure A.5: Polar domain - Impact of baseline rankers of various quality levels at times $t \in \{1, 2, 10\}$.

## A.2 Aspect Modeling

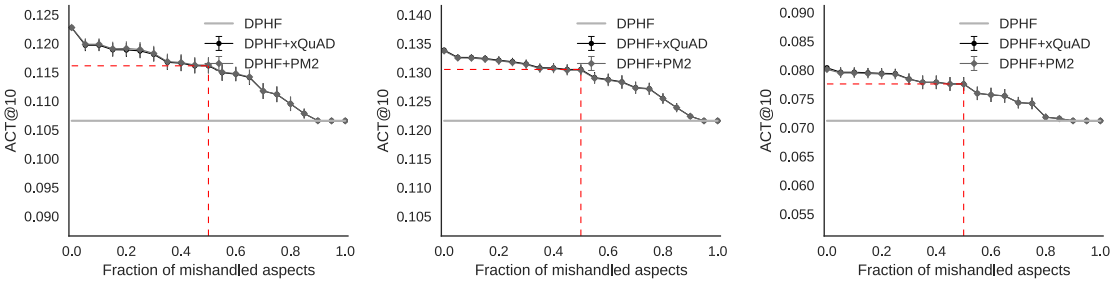



(a) Ebola 2015 (b) Local Politics (c) Illicit Goods

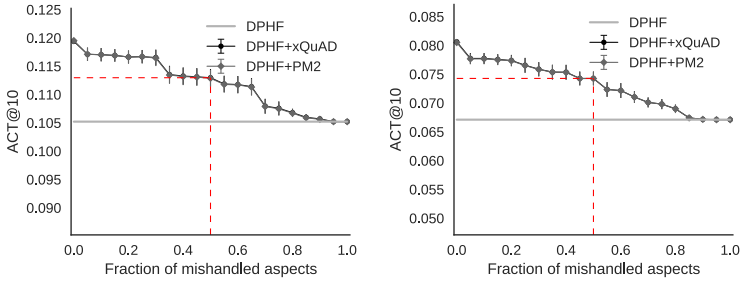



(d) Ebola 2016 (e) Polar

Figure A.6: Inaccurate aspect modeling.



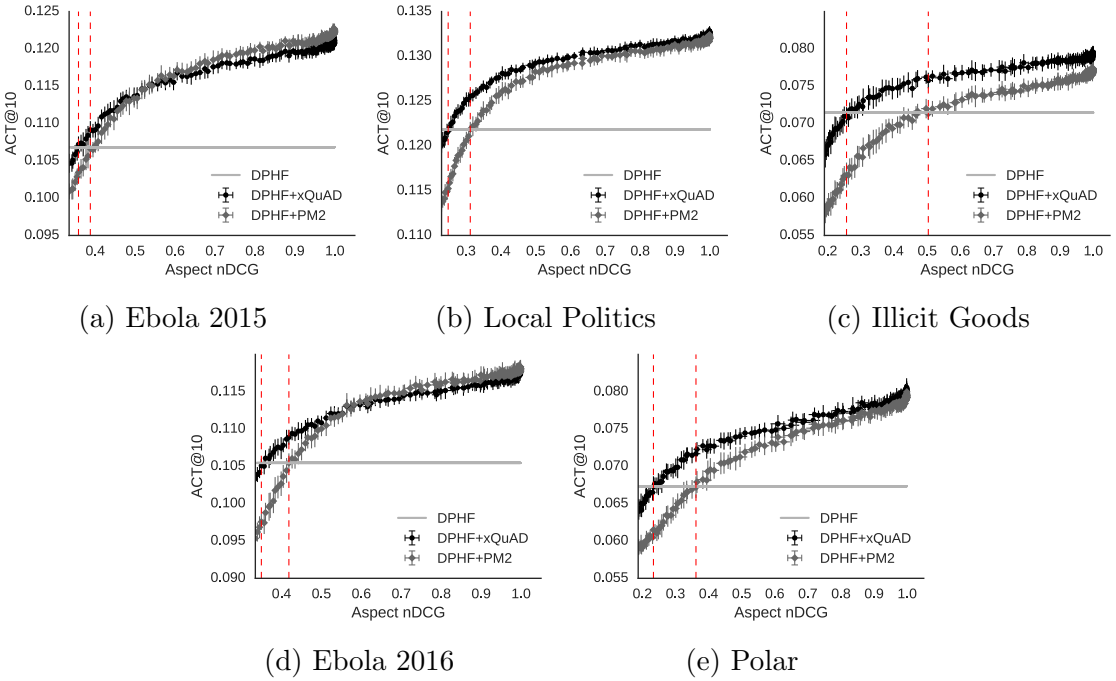

(a) Ebola 2015 (b) Local Politics (c) Illicit Goods




(d) Ebola 2016 (e) Polar

Figure A.7: Incomplete aspect modeling.

## A.3 Dynamic Reranker




(a) Ebola 2015 (b) Local Politics (c) Illicit Goods

(d) Ebola 2016 (e) Polar

Figure A.8: Impact of perturbed coverage estimates.

# Appendix B

# Practical Instantiations

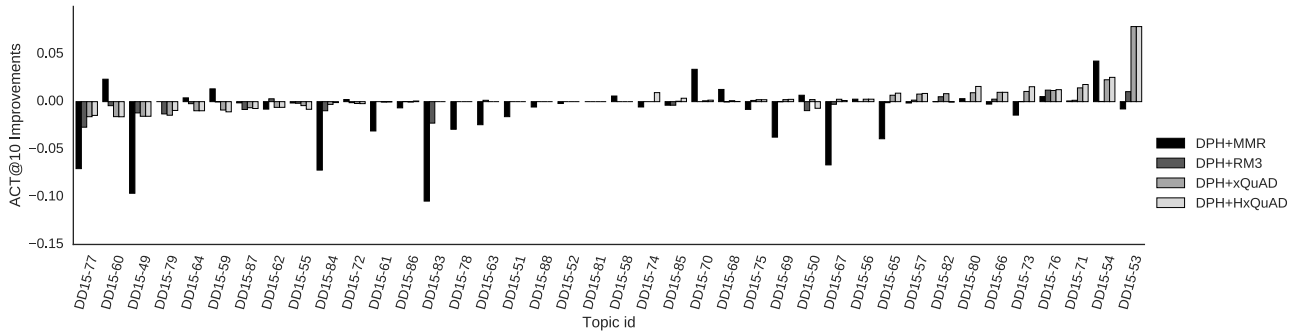



Figure B.1: Ebola 2015

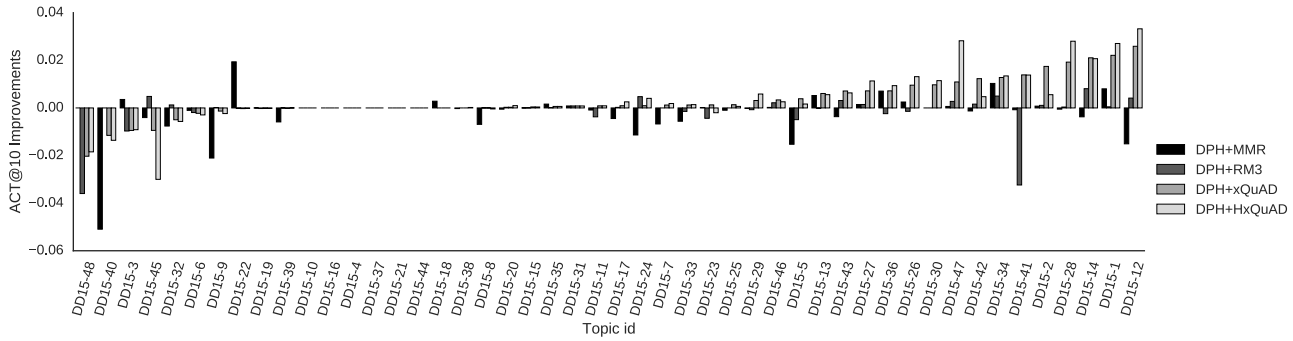



Figure B.2: Local Politics

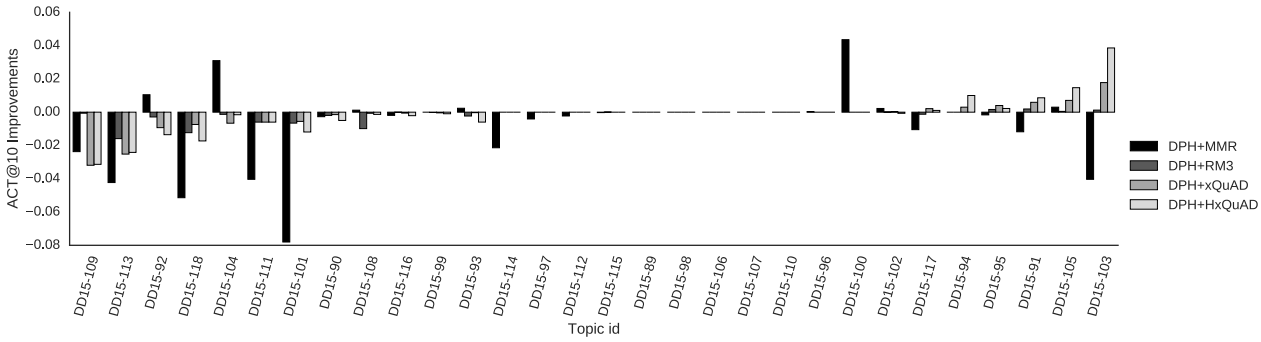



Figure B.3: Illicit Goods

# Bibliography

Agrawal, R., Gollapudi, S., Halverson, A., and Ieong, S. (2009). Diversifying search results. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, WSDM '09, pages 5--14, New York, NY, USA. ACM.

Allan, J. (2006). Hard track overview in trec 2005: High accuracy retrieval from documents. In *Proceedings of the 14th Text REtrieval Conference*.

Amati, G., Ambrosi, E., Bianchi, M., Gaibisso, C., and Gambosi, G. (2007). Fub, iasi-cnr and university of tor vergata at trec 2007 blog track. In *Proceedings of the 16th Text REtrieval Conference*.

Athukorala, K., Gowacka, D., Jacucci, G., Oulasvirta, A., and Vreeken, J. (2016). Is exploratory search different? a comparison of information search behavior for exploratory and lookup tasks. *Journal of the Association for Information Science and Technology*, 67(11):2635–2651.

Baeza-Yates, R. A. and Ribeiro-Neto, B. A. (2011). *Modern Information Retrieval - the concepts and technology behind search, Second edition*. Pearson Education Ltd., Harlow, England. ISBN 978-0-321-41691-9.

Barbosa, L. and Freire, J. (2007). An adaptive crawler for locating hidden-web entry points. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 441--450, New York, NY, USA. ACM.

Bendersky, M., Metzler, D., and Croft, W. B. (2010). Learning concept importance using a weighted dependence model. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, WSDM '10, pages 31--40, New York, NY, USA. ACM.

Bendersky, M., Metzler, D., and Croft, W. B. (2011). Parameterized concept weighting in verbose queries. In *Proceedings of the 34th International ACM SIGIR Conference*

*on Research and Development in Information Retrieval*, SIGIR '11, pages 605--614, New York, NY, USA. ACM.

Bendersky, M., Metzler, D., and Croft, W. B. (2012). Effective query formulation with multiple information sources. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, WSDM '12, pages 443--452, New York, NY, USA. ACM.

Berlioz, A., Friedman, A., Kaafar, M. A., Boreli, R., and Berkovsky, S. (2015). Applying differential privacy to matrix factorization. In *Proceedings of the 9th ACM Conference on Recommender Systems*, RecSys '15, pages 107--114, New York, NY, USA. ACM.

Carbonell, J. and Goldstein, J. (1998). 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*, SIGIR '98, pages 335--336, New York, NY, USA. ACM.

Carterette, B., Clough, P., Hall, M., Kanoulas, E., and Sanderson, M. (2016). Evaluating retrieval over sessions: The trec session track 2011-2014. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, pages 685--688, New York, NY, USA. ACM.

Chapelle, O., Ji, S., Liao, C., Velipasaoglu, E., Lai, L., and Wu, S.-L. (2011). Intent-based diversification of web search results: Metrics and algorithms. *Inf. Retr.*, 14(6):572--592. ISSN 1386-4564.

Chapelle, O., Metlzer, D., Zhang, Y., and Grinspan, P. (2009). Expected reciprocal rank for graded relevance. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 621--630, New York, NY, USA. ACM.

Clarke, C. L., Kolla, M., Cormack, G. V., Vechtomova, O., Ashkan, A., Büttcher, S., and MacKinnon, I. (2008). Novelty and diversity in information retrieval evaluation. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pages 659--666, New York, NY, USA. ACM.

Clarke, C. L. A. and Smucker, M. D. (2014). Time well spent. In *Proceedings of the 5th Information Interaction in Context Symposium*, IIiX '14, pages 205--214, New York, NY, USA. ACM.

Cooper, W. S. (1997). Readings in information retrieval. chapter On Selecting a Measure of Retrieval Effectiveness. Part I., pages 191--204. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Dang, V. and Croft, W. B. (2012). Diversity by proportionality: An election-based approach to search result diversification. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, pages 65--74, New York, NY, USA. ACM.

Ferro, N., Silvello, G., Keskustalo, H., Pirkola, A., and Järvelin, K. (2016). The twist measure for IR evaluation: Taking user's effort into account. *JASIST*, 67(3):620--648.

Golovchinsky, G., Diriye, A., and Dunnigan, T. (2012). The future is in the past: Designing for exploratory search. In *Proceedings of the 4th Information Interaction in Context Symposium*, IIIX '12, pages 52--61, New York, NY, USA. ACM.

Grace Hui Yang, I. S. (2015). Trec 2015 dynamic domain track overview. In *Proceedings of the 24th Text REtrieval Conference*, Gaithersburg, MD, USA.

Grace Hui Yang, I. S. (2016). Trec 2016 dynamic domain track overview. In *Proceedings of the 25th Text REtrieval Conference*, Gaithersburg, MD, USA.

Hu, S., Dou, Z., Wang, X., Sakai, T., and Wen, J.-R. (2015). Search result diversification based on hierarchical intents. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, pages 63--72, New York, NY, USA. ACM.

Jiang, J. and Allan, J. (2016). Adaptive effort for search evaluation metrics. In *Advances in Information Retrieval - 38th European Conference on IR Research, ECIR 2016, Padua, Italy, March 20-23, 2016. Proceedings*, pages 187--199.

Jiang, J., He, D., Kelly, D., and Allan, J. (2017). Understanding ephemeral state of relevance. In *Proceedings of the 2016 ACM on Conference on Human Information Interaction and Retrieval*, CHIIR '17, pages 111--120, New York, NY, USA. ACM.

Jin, X., Sloan, M., and Wang, J. (2013). Interactive exploratory search for multi page search results. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 655--666, New York, NY, USA. ACM.

Kanoulas, E. (2016). *A Short Survey on Online and Offline Methods for Search Quality Evaluation*, pages 38--87. Springer International Publishing, Cham.

Krishnamurthy, Y., Pham, K., Santos, A., and Freire, J. (2016). Interactive exploration for domain discovery on the web. In *KDD 2016 Workshop on Interactive Data Exploration and Analytics*, IDEA '16.

Lavrenko, V. and Croft, W. B. (2001). Relevance based language models. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, pages 120--127, New York, NY, USA. ACM.

Li, X. and Zhu, Z. (2008). Enhancing relevance models with adaptive passage retrieval. In *Proceedings of the IR Research, 30th European Conference on Advances in Information Retrieval*, ECIR'08, pages 463--471, Berlin, Heidelberg. Springer-Verlag.

Liu, T.-Y. (2009). Learning to rank for information retrieval. *Found. Trends Inf. Retr.*, 3(3):225--331. ISSN 1554-0669.

Lloyd, S. (2006). Least squares quantization in pcm. *IEEE Trans. Inf. Theor.*

Luo, C., Liu, Y., Sakai, T., Zhou, K., Zhang, F., Li, X., and Ma, S. (2017). Does document relevance affect the searcher's perception of time? In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, WSDM '17, pages 141--150, New York, NY, USA. ACM.

Luo, J., Dong, X., and Yang, H. (2015a). Learning to reinforce search effectiveness. In *Proceedings of the 2015 International Conference on The Theory of Information Retrieval*, ICTIR '15, pages 271--280, New York, NY, USA. ACM.

Luo, J., Dong, X., and Yang, H. (2015b). Session search by direct policy learning. In *Proceedings of the 2015 International Conference on The Theory of Information Retrieval*, ICTIR '15, pages 261--270, New York, NY, USA. ACM.

Luo, J., Wing, C., Yang, H., and Hearst, M. (2013). The water filling model and the cube test: Multi-dimensional evaluation for professional search. In *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*, CIKM '13, pages 709--714, New York, NY, USA. ACM.

Luo, J., Zhang, S., and Yang, H. (2014). Win-win search: Dual-agent stochastic game in session search. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '14, pages 587--596, New York, NY, USA. ACM.

Marchionini, G. (2006). Exploratory search: From finding to understanding. *Commun. ACM*, 49(4):41--46. ISSN 0001-0782.

Maxwell, D., Azzopardi, L., Järvelin, K., and Keskustalo, H. (2015). Searching and stopping: An analysis of stopping rules and strategies. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, pages 313--322, New York, NY, USA. ACM.

Metzler, D. and Bruce Croft, W. (2007). Linear feature-based models for information retrieval. *Inf. Retr.*, 10(3):257--274. ISSN 1386-4564.

Moffat, A. and Zobel, J. (2008). Rank-biased precision for measurement of retrieval effectiveness. *ACM Trans. Inf. Syst.*, 27(1):2:1--2:27. ISSN 1046-8188.

Moraes, F., Santos, R. L. T., and Ziviani, N. (2016). UFMG at the TREC 2016 Dynamic Domain track. In *Proceedings of the 25th Text REtrieval Conference*, Gaithersburg, MD, USA.

Na, S.-H., Kang, I.-S., Lee, Y.-H., and Lee, J.-H. (2008). Applying completely-arbitrary passage for pseudo-relevance feedback in language modeling approach. In *Information Retrieval Technology, 4th Asia Infomation Retrieval Symposium, AIRS 2008, Harbin, China, January 15-18, 2008, Revised Selected Papers*, pages 626--631.

Nemhauser, G. L., Wolsey, L. A., and Fisher, M. L. (1978). An analysis of approximations for maximizing submodular set functions—i. *Mathematical Programming*.

Odijk, D., Meij, E., Sijaranamual, I., and de Rijke, M. (2015). Dynamic query modeling for related content finding. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, pages 33--42, New York, NY, USA. ACM.

Robertson, S., Walker, S., Jones, S., Hancock-Beaulieu, M., and Gatford, M. (1996). Okapi at trec-3. In *Overview of TREC-3*.

Rocchio, J. J. (1971). Relevance feedback in information retrieval. In Salton, G., editor, *The Smart retrieval system - experiments in automatic document processing*, pages 313--323. Englewood Cliffs, NJ: Prentice-Hall.

Ruotsalo, T., Jacucci, G., Myllymäki, P., and Kaski, S. (2014). Interactive intent modeling: Information discovery beyond search. volume 58, pages 86--92, New York, NY, USA. ACM. ISSN 0001-0782.

Sakai, T. and Dou, Z. (2013). Summaries, ranked retrieval and sessions: A unified framework for information access evaluation. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 473--482, New York, NY, USA. ACM.

Salton, G., Wong, A., and Yang, C. S. (1975). A vector space model for automatic indexing. *Commun. ACM*.

Sanderson, M. (2010). Test collection based evaluation of information retrieval systems. *Foundations and Trends in Information Retrieval*, 4(4):247–375. ISSN 1554-0669.

Santos, R. L., Macdonald, C., and Ounis, I. (2010). Exploiting query reformulations for web search result diversification. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 881--890, New York, NY, USA. ACM.

Santos, R. L. T., Macdonald, C., and Ounis, I. (2012). On the role of novelty for search result diversification. *Information Retrieval*.

Santos, R. L. T., Macdonald, C., and Ounis, I. (2015). *Search result diversification*. Now Publishers.

Saracevic, T. (1975). RELEVANCE: A review of and a framework for the thinking on the notion in information science. *JASIS*, 26(6):321--343.

Saracevic, T. (2007a). Relevance: A review of the literature and a framework for thinking on the notion in information science. part II: nature and manifestations of relevance. *JASIST*, 58(13):1915--1933.

Saracevic, T. (2007b). Relevance: A review of the literature and a framework for thinking on the notion in information science. part III: behavior and effects of relevance. *JASIST*, 58(13):2126--2144.

Sloan, M. and Wang, J. (2015). Dynamic information retrieval: Theoretical framework and application. In *Proceedings of the 2015 International Conference on The Theory of Information Retrieval*, ICTIR '15, pages 61--70, New York, NY, USA. ACM.

Smucker, M. D. and Clarke, C. L. (2012). Time-based calibration of effectiveness measures. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, pages 95--104, New York, NY, USA. ACM.

Turpin, A. and Scholer, F. (2006). User performance versus precision measures for simple search tasks. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 11--18, New York, NY, USA. ACM.

Verma, M., Yilmaz, E., and Craswell, N. (2016). On obtaining effort based judgements for information retrieval. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, WSDM '16, pages 277--286, New York, NY, USA. ACM.

White, R. W. (2016). *Interactions with Search Systems*. Cambridge University Press.

Wildemuth, B. M. and Freund, L. (2012). Assigning search tasks designed to elicit exploratory search behaviors. In *Proceedings of the Symposium on Human-Computer Interaction and Information Retrieval*, HCIR '12, pages 4:1--4:10, New York, NY, USA. ACM.

Xu, Y. C. and Chen, Z. (2006). Relevance judgment: What do information users consider beyond topicality? *J. Am. Soc. Inf. Sci. Technol.*, 57(7):961--973. ISSN 1532-2882.

Yang, G. H., Sloan, M., and Wang, J. (2016). *Dynamic Information Retrieval Modeling*. Morgan & Claypool Publishers.

Yilmaz, E., Verma, M., Craswell, N., Radlinski, F., and Bailey, P. (2014). Relevance and effort: An analysis of document utility. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, CIKM '14, pages 91--100, New York, NY, USA. ACM.

Yilmaz, E., Verma, M., Mehrotra, R., Kanoulas, E., Carterette, B., and Craswell, N. (2015). Overview of the TREC 2015 tasks track. In *Proceedings of the 25th Text REtrieval Conference*.

Yue, Y. and Joachims, T. (2009). Interactively optimizing information retrieval systems as a dueling bandits problem. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 1201--1208, New York, NY, USA. ACM.

Zhai, C. (2008). Statistical language models for information retrieval a critical review. *Foundations and Trends in Information Retrieval*.

Zhai, C. and Lafferty, J. (2001). Model-based feedback in the language modeling approach to information retrieval. In *Proceedings of the Tenth International Conference on Information and Knowledge Management*, CIKM '01, pages 403--410, New York, NY, USA. ACM.