ADVANCED METHODS FOR DIVERSIFICATION OF RESULTS IN GENERAL-PURPOSE AND SPECIALIZED SEARCH ENGINES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

SEVGİ YİĞİT SERT

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
COMPUTER ENGINEERING

DECEMBER 2020

Approval of the thesis:

ADVANCED METHODS FOR DIVERSIFICATION OF RESULTS IN GENERAL-PURPOSE AND SPECIALIZED SEARCH ENGINES

submitted by **SEVGİ YİĞİT SERT** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. Halit Oğuztüzün
Head of Department, **Computer Engineering** _____

Assoc. Prof. Dr. İsmail Sengör Altıngövde
Supervisor, **Computer Engineering, METU** _____

Prof. Dr. Özgür Ulusoy
Co-supervisor, **Computer Engineering, Bilkent University** _____

Examining Committee Members:

Prof. Dr. İsmail Hakkı Toroslu
Computer Engineering, METU _____

Assoc. Prof. Dr. İsmail Sengör Altıngövde
Computer Engineering, METU _____

Prof. Dr. Pınar Karagöz
Computer Engineering, METU _____

Prof. Dr. Fazlı Can
Computer Engineering, Bilkent University _____

Assist. Prof. Dr. Engin Demir
Computer Engineering, Hacettepe University _____

Date: 28.12.2020

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Sevgi Yiğit Sert

Signature :

ABSTRACT

ADVANCED METHODS FOR DIVERSIFICATION OF RESULTS IN GENERAL-PURPOSE AND SPECIALIZED SEARCH ENGINES

Yiğit Sert, Sevgi
Ph.D., Department of Computer Engineering
Supervisor: Assoc. Prof. Dr. İsmail Sengör Altıngövde
Co-Supervisor: Prof. Dr. Özgür Ulusoy

December 2020, 129 pages

Diversifying search results is a common mechanism in information retrieval to satisfy more users by surfacing documents that address different possible intentions of users. It aims to generate a result list that is both relevant and diverse when ambiguous and/or broad queries appear. Such queries have different underlying subtopics (a.k.a., aspects or interpretations) that search result diversification algorithms should consider. In this thesis, we first address search result diversification as a useful method to support search as learning, since diversification ensures to cover all possible aspects of the query in the final ranking. We argue that, in a search engine for the education domain, it is appropriate to diversify results across multiple dimensions, including the suitability of the content for different education levels and the type of the document in addition to topical ambiguity. We introduce a framework that extends the probabilistic and supervised methods for diversification that can consider the aspects of multiple independent dimensions during ranking, and demonstrate its effectiveness on a newly developed test collection.

As our second contribution, we propose three different frameworks that exploit super-

vised learning methods to improve the effectiveness of explicit search result diversification, which presumes that query aspects are known during diversification. We also, for the first time in the literature, propose to learn the importance of aspects by leveraging query performance predictors (QPPs). We conduct our exhaustive experiments on a commonly used benchmark dataset and show that explicit diversification performance can be considerably improved using supervised learning methods without requiring large training sets or high computing capabilities.

As a third contribution of this thesis, we examine the impact of static index pruning on diversification performance. We introduce two novel strategies that take into account the topical diversity of documents and preserve documents relevant to different aspects while pruning the index. We show that our proposed pruning strategies outperform the existing approaches in terms of various diversification measures.

Keywords: Explicit search result diversification, search as learning, supervised learning, query performance predictors, static index pruning.

ÖZ

GENEL-AMAÇLI VE ÖZELLEŞMİŞ ARAMA MOTORLARINDA SONUÇ ÇEŞİTLENDİRME İÇİN İLERİ YÖNTEMLER

Yiğit Sert, Sevgi
Doktora, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Doç. Dr. İsmail Sengör Altıngövde
Ortak Tez Yöneticisi: Prof. Dr. Özgür Ulusoy

Aralık 2020 , 129 sayfa

Arama sonuçlarının çeşitlendirilmesi, kullanıcıların olası farklı isteklerine hitap eden belgeleri ortaya çıkararak daha fazla kullanıcıyı memnun etmek için kullanılan yaygın bir bilgi erişim mekanizmasıdır. Belirsiz ve kapsamlı sorgular için mümkün olduğunca hem ilgili hem de çeşitli bir sonuç listesi oluşturulması amaçlanır. Bu sorgular, arama sonucu çeşitlendirme algoritmalarının dikkate alması gereken farklı alt konulara (diğer bir deyişle yönlere) sahiptir. Bu tezde, ilk olarak, sorguya cevap olarak önerilen sonuç sıralamasında sorgunun tüm olası yönlerini kapsamayı sağladığından, arama sonucu çeşitlendirilmesi ararken öğrenmeyi destekleyen yararlı bir yöntem olarak ele alınmaktadır. Eğitsel bir arama motorunda, konuya ilişkin belirsizliğe ek olarak belge içeriğinin farklı eğitim düzeylerine uygunluğu ve belgenin türü dahil olmak üzere farklı birden çok boyutta arama sonuçlarının çeşitlendirilmesinin uygun olduğu tartışılmaktadır. Olasılıksal yöntemlere ve denetimli öğrenmeye dayanan cevap çeşitlendirme yöntemlerini, sıralama sırasında birden çok bağımsız boyutun alt konularını dikkate alabilecek şekilde genişleten bir çerçeve sunulmakta ve yeni geliştirilen test

koleksiyonunda önerilen yöntemlerin etkinliği gösterilmektedir.

Bu tezin ikinci özgün katkısı olarak, sorgu alt konularının çeşitlendirme sırasında bilindiğini varsayan dolaysız arama sonucu çeşitlendirmesinin etkinliğini artırmak için denetimli öğrenme yöntemlerinden yararlanan üç farklı çerçeve önerilmektedir. Ayrıca, literatürde ilk kez, sorgu başarım tahmincilerinden (SBT'ler) yararlanarak sorgu alt konularının öneminin öğrenilmesi önerilmektedir. Kapsamlı deneylerimizi yaygın olarak kullanılan bir karşılaştırmalı değerlendirme veri kümesi üzerinde gerçekleştirmekte ve dolaysız arama sonucu çeşitlendirme performansının, büyük eğitim kümeleri veya yüksek hesaplama kapasitesi gerektirmeden denetimli öğrenme yöntemleri kullanılarak önemli ölçüde iyileştirilebildiğini göstermekteyiz.

Son olarak, statik indeks budamanın çeşitlendirme performansı üzerindeki etkisi incelenmektedir. Belgelerin konusal çeşitliliğini hesaba katan ve indeksi budarken farklı alt konulara sahip belgeleri koruyan iki yeni strateji sunulmaktadır. Önerilen budama stratejilerimizin çeşitlendirme etkinliği açısından mevcut yaklaşımlardan daha iyi performans sergilediği gösterilmektedir.

Anahtar Kelimeler: Dolaysız arama sonuçlarını çeşitlendirme, ararken öğrenme, denetimli öğrenme, sorgu başarım tahmincileri, statik indeks budama.

To my family

ACKNOWLEDGMENTS

I would like to thank sincerely my supervisor Assoc. Prof. Dr. İsmail Sengör Altıngövde who reminds me repeatedly that the most valuable thing is to research, read, question, and never go easy. Throughout my Ph.D. studies, he guided and helped me through all the problems I faced with his invaluable knowledge and contributions. I also would like to express my sincere gratitude to my co-supervisor Prof. Dr. Özgür Ulusoy for his guidance, support, motivation, and patience.

I wish to thank Prof. Dr. Pınar Karagöz and Assist. Prof. Dr. Engin Demir, who were in my thesis monitoring committee, for their constructive feedback; and also to thank Prof. Dr. Fazlı Can and Prof. Dr. İsmail Hakkı Toroslu who participated in my defense jury for their valuable comments. My special thanks go to Prof. Dr. Iadh Ounis and Dr. Craig MacDonald from the School of Computing Science at the University of Glasgow for their guidance and helpful suggestions during this work.

I would like to thank The Scientific and Technological Research Council of Turkey (TÜBİTAK) for supporting me by Ph.D. Scholarship Program (support type 2211-A). I also thank TÜBİTAK (grant no. 117E861) and the Royal Society under the Newton Int.'l Exchanges Scheme (grant no. NI140231) for partially funding our work (i.e., for research visits and equipment costs).

I would like to thank my dear friends Pınar Küllü and Merve Özkan Okay, who has been with me with all their sincerity and friendship at every moment I need; also thank my colleagues, Yılmaz Ar, and Gazi Erkan Bostancı. I also thank to Daniel Sheldon for providing the source code of LambdaMerge.

I must express my deepest gratitude to my mother and father for their encouragement, unconditional support, and eternal love and patience throughout my life. Finally, I'm grateful to my husband Uğur Sert who made me feel relieved and motivated at every despairing moment during my Ph.D. journey. Without his patience and all kinds of support, this thesis would have never been completed.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ALGORITHMS

ALGORITHMS

LIST OF ABBREVIATIONS

ABBREVATIONS	
IR	Information Retrieval
xQuAD	Explicit Query Aspect Diversification
LTR	Learning-To-Rank
R-LTR	Relational Learning To Rank
IA-Select	Intent-Aware Select
TREC	Text Retrieval Conference
ERR-IA	Intent-Aware Expected Reciprocal Rank
DCG	Discounted Cumulative Gain
nDCG	Normalized Discounted Cumulative Gain
α-nDCG	α weighted normalized Discounted Cumulative Gain
P-IA	Intent-Aware Precision
iSEEK	Internet System for Education and Employment Knowledge
TIB	German National Library of Science and Technology
MMR	Maximum Marginal Relevance
MMRE	MMR-based Expansion
SGD	Stochastic Gradient Descent
WWW	World Wide Web
QPP	Query Performance Predictor
SVM	Support Vector Machine
NTN	Neural Tensor Network
PP	Popularity-based Pruning
aTCP	Access-based Term-Centric Pruning
aDCP	Access-based Document-Centric Pruning

IDF	Inverse Document Frequency
RIDF	Residual Inverse Document Frequency
PRP	Probability Ranking Principle
KLD	Kullback-Leibler Divergence
URL	Uniform Resource Locator

CHAPTER 1

INTRODUCTION

1.1 Motivation

Information Retrieval (IR) systems usually aim to find a ranked list of documents within large collections that contain desired information in response to a user query. The documents in these collections are inherently unstructured (commonly composed of text) [2]. Furthermore, as the World Wide Web (WWW) grows and evolves so rapidly, surfacing the documents that a user seeks from the massive amount of data becomes more compelling.

Web search engines come into play here and provide a connection between users and information on the WWW. Users interact with search engines by typing a few keywords, called a *query*, to express their information need. User queries are mostly short, i.e. more than half of them are one or two terms [3] that make it harder to identify the user information need. Moreover, these queries are often ambiguous - have more than one interpretation- or underspecified - have multiple aspects of the query. For instance, *java* is an ambiguous query. The user may search for information about different aspects (a.k.a., intents or sub-topics) of the query *java* such as an island in Indonesia, a type of coffee, or a computer programming language. Even if the user is known to be a computer enthusiast, the query *java* is not clear enough to precisely understand the user's information need. The user may want to know about *how to build java*, or *java development environments* or *how to learn java coding*, which are different aspects of the query.

As the predominant purpose of search engines is users' satisfaction, they should represent their search results in a diversified manner in addition to providing relevant

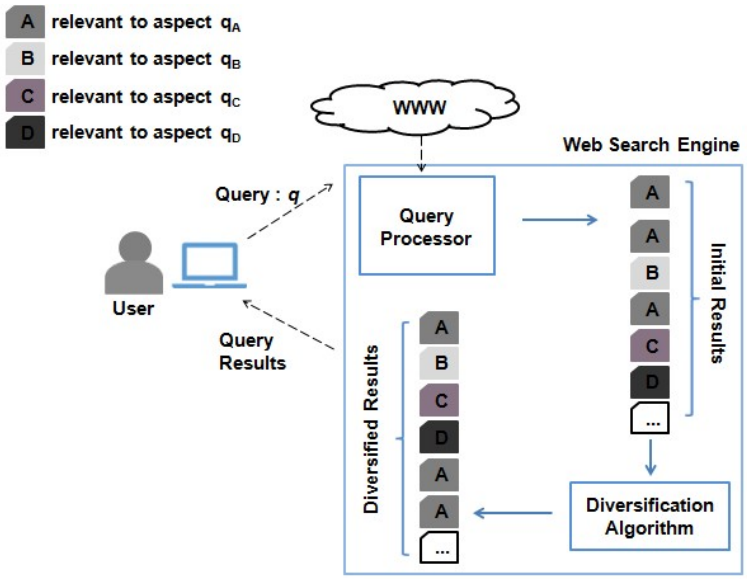


Figure 1.1: The diversification procedure.

results to the query. Thus, users who type the same query but have diverse information needs can be satisfied by seeing at least one relevant result.

Search result diversification aims to bring results with a wide coverage of all possible aspects/subtopics (they are interchangeably used throughout this thesis) to improve user satisfaction when there is no available information about the user. It consists of two stages. First, documents are sorted according to their relevance to the given query by a ranking algorithm. Next, the initial retrieval results, which comprise of top-N documents in the ranked list, are re-ranked by a diversification algorithm to incorporate diversity. This procedure is illustrated in Figure 1.1.

Existing approaches in the literature for search result diversification can be defined as implicit and explicit, with respect to how they address the aspects of a query. The implicit approaches make use of differences between documents in the initially retrieved set assuming that there exists no prior knowledge about the aspects of the query. As for explicit approaches, they assume the explicit availability of query aspects which can be modeled by using an external source such as a taxonomy [4] or query log [5]. They usually consider the coverage of these aspects to choose the next document into the diversified ranking.

Recently, search result diversification has been regarded as a crucial topic, and takes

lots of attention from researchers [6, 7, 8, 9]. In addition to satisfying users with different intents, it also helps the user to learn more about a topic since it allows presenting different aspects of the topic for a given query. From this point of view, diversification becomes even more important as it supports learning during the search process [10, 11]. Furthermore, result diversification methods promise to mitigate search bias [12] and enhance fairness in various search scenarios [13], where it is reasonable to assume the availability of such query aspects (such as gender, ethnicity, and age in a job search scenario [14]).

In this thesis, we propose effective and robust strategies for search result diversification that are applicable for general and/or specialized search engines.

1.2 Contributions

The contributions of this thesis comprise of devising effective techniques for search result diversification problem from different viewpoints. First, in Chapter 2, we address search result diversification in the context of educational search, because diversifying search results has been emerging as a useful technique to learn a topic or to gain deeper knowledge about a certain topic as it ensures surfacing various aspects of the topic. As the first contribution, we show that, in an educational search engine, diversifying search results over for just topical aspects of retrieved documents is inadequate, it is necessary to diversify across multiple dimensions to support learning. Next, we propose a novel framework that recast the well-known probabilistic and supervised diversification algorithms (i.e., xQuAD [15], a variant of PM2 [16] and R-LTR [17]) to take into consideration multiple dimensions. We also enhance each method by incorporating a new component, namely dimension importance, into the computation of coverage of query aspects. We introduce a new rich dataset specifically developed to facilitate the evaluation of diversification algorithms with multiple dimensions in the context of a deployed educational search engine. This dataset is obtained from user interactions with a real-world educational search engine. Thus, for the performance analysis of our work, we use a more realistic set-up which is based on query instances and clicks in addition to TREC-style relevance annotations[1] [18].

[1] https://trec.nist.gov/

Experiments show that the proposed approach is effective, as we obtain substantial improvements in terms of several diversification metrics (e.g., 2.4%, 0.93%, and 2% for ERR-IA [19], α-nDCG [20] and P-IA [4], respectively; and 1.4% for the traditional P@2 metric). This work was published in:

- Yigit-Sert, S., Altingovde, I. S., Macdonald, C., Ounis, I., & Ulusoy, Ö. Explicit diversification of search results across multiple dimensions for educational search. Journal of the Association for Information Science and Technology. https://doi.org/10.1002/asi.24403.

In Chapter 3, we employ supervised learning methods to improve diversification performance in three different frameworks with different features and goals. Our first framework, LTRDiv, enables us to apply typical LTR algorithms for diversification context in an intuitive way. To the best of our knowledge, it is the first work that casts the diversification problem as that of learning a ranking model, which is based on the aggregated relevance of each document to all aspects of a given query. Our proposed approach is motivated by the observation that, theoretically, a ranking where documents are sorted with respect to the number of aspects they are relevant to (i.e., *covered* aspects) will maximize the well-known intent-aware diversity metrics (under certain assumptions that will be discussed later), and such a ranking is also likely to satisfy the users' diversity requirements in practice. Therefore, we train models to predict the number of covered aspects *per document*. With our second framework, AspectRanker, for the first time in the literature, we propose to *learn* the importance of aspects by re-ranking the candidate set for each aspect and leveraging query performance predictors (QPPs). In the closest work to ours in the literature, [21] directly employed such QPP estimates as the aspect importance values, but they did not explore the idea of training a model to predict these values, as we do in the AspectRanker framework. Lastly, we cast the diversification problem as a fusion task, namely the supervised merging of rankings per query aspect. To this end, we adopt an existing approach, LambdaMerge [1], to our problem. Again, earlier works only considered traditional (unsupervised) merging strategies in this context (e.g., [22]), but did not exploit supervised learning. In our adaptation of LambdaMerge, we learn the aspects' importance based on the representation quality of each aspect in the *can-*

didate set (as in AspectRanker), and we optimize an objective function that takes into account the relevance (of a document) to multiple aspects of a query.

We conduct experiments using the topic sets introduced in the Diversity task of the TREC Web Track between 2009 and 2012. For each topic set, we first generate an initial retrieval result (i.e., *run*) based on the BM25 matching function, which is a traditional function and still employed in practical systems. Additionally, our experiments also use the best-performing run submitted to the ad-hoc retrieval track of TREC in each year. The latter setup allows to demonstrate the robustness of our methods, as these runs usually employ sophisticated retrieval methods beyond BM25. We show that our proposed frameworks, especially AspectRanker and LmDiv, outperform two strong baselines and reveal that diversification performance can be enhanced without using large training sets and high computational cost. This chapter was published in:

- Yigit-Sert, S., Altingovde, I. S., Macdonald, C., Ounis, I.,& Ulusoy, Ö. (2020). Supervised approaches for explicit search result diversification. Information Processing and Management, 57(6), 102356.

In Chapter 4, we investigate the diversification effectiveness of retrieved results when a static index pruning strategy is applied over the underlying inverted index. We demonstrate that the diversity of the top-ranked results is seriously hurt especially when the index is pruned aggressively (i.e., from 60% up to 90%). As a remedy, we propose two new strategies that take into account the topical diversity of documents while pruning. Specifically, we cluster the documents regarding their contents and and apply alternative methods to keep in the index the documents that pertain to different aspects of a topic.

We conclude this thesis with a brief summary and discuss possible future work directions in Chapter 5.

CHAPTER 2

EXPLICIT DIVERSIFICATION OF SEARCH RESULTS ACROSS MULTIPLE DIMENSIONS

Diversifying search results is a common technique to address queries where there is some topical ambiguity in exactly what relevant documents will address the users' needs. By surfacing diverse content within the top-ranked results, multiple users with differing information needs can be satisfied. Nowadays result diversification is also embraced as a technique in contributing users' search and learning experiences by representing a broader view on the topic. In this chapter[1], we argue that, in a search engine for the education domain, it is appropriate to diversify documents across multiple dimensions such as the type of the document (e.g., text, video, animation, etc.) or its education level. We propose a framework that extends the probabilistic and supervised models for diversification across multiple dimensions.

In Section 2.1, we define our research problem, multi-dimensional result diversification, in the context of educational search. In Section 2.2, we review the related works that focus on the *search as learning* paradigm in the context of general-purpose and educational search engines, and then position our work in the search result diversification literature as a whole. Section 2.3 describes our models for diversification across multiple dimensions as adaptations of the xQuAD, PM2 and R-LTR methods, and provides a particular instantiation of these models in the context of an educational search engine. We present the search evaluation dataset developed for this work and the evaluation results in Sections 2.4 and 2.5, respectively. In the last section, we provide concluding remarks.

[1] Reprinted by permission with license number 4954090494924 from The Journal of the Association for Information Science and Technology (JASIST), S. Yigit-Sert, I. S. Altingovde, C. Macdonald, I. Ounis, Ö. Ulusoy, Explicit diversification of search results across multiple dimensions for educational search, ©2020, John Wiley and Sons. https://doi.org/10.1002/asi.24403.

2.1 Introduction

Exploiting search as a process for learning is an emerging and exciting research direction (a.k.a. *search as learning*) that attracts interest from various fields, such as computer science, psychology and learning sciences [10, 23, 24]. A particular direction in recent studies, especially from the perspective of information retrieval research, has addressed how general-purpose search engines can be exploited and enriched to satisfy the users' possible learning goals. To this end, earlier works attempted to re-rank the results of a search engine by applying various techniques, most remarkably, personalization or diversification. For instance, some works [25, 26] personalized the displayed ranking by incorporating the reading difficulty of documents. In contrast, Raman et al. [27] proposed a method to diversify the retrieved document set in terms of the different topical aspects for the so-called exploratory queries. Inspired by the latter work, we propose applying diversification across *multiple-dimensions* in the context of *educational search* (more specifically, for a search engine that is built on educational resources and primarily accessed by students and teachers).

While the aforementioned previous works paved the way for improving educational search, they essentially focused on a single-dimension, e.g., either the topical aspect or the reading difficulty, for leveraging during the re-ranking of query results. Instead, in this study, we argue that search activity for learning can benefit from diversity in the result list — as also suggested in the earlier works [28] – yet the diversity should better be provided for *multiple-dimensions*. That is, the result list should not only be diversified for the topical aspects covered by the retrieved documents, but also for other dimensions, such as the type of the document (e.g., text, video, animation, or even a test to assess what has been learnt) or its intellectual level (say, for beginners or experts). Our standpoint is addressing a need that has been recognized also by others. For instance, Hoppe et al. [24] identify the 'lack of consideration for multimodal resources' as a major challenge for the search as learning paradigm. Furthermore, even for a fixed topical aspect, it could still be possible and preferable to provide information at various levels, say, at a beginner level with some basics, or at the expert level with advanced content, so that users with different levels of initial knowledge can still benefit from the search (and advance their knowledge gradually, by accessing the

results at different levels). Hence, we propose diversification in multiple-dimensions to obtain a re-ranking of results that can complement learning via search in multiple ways (i.e., presentation of information at alternative *forms* and *levels*), beyond coverage of topical variety.

Take an illustrative example query, 'triangle', which may have underlying aspects such as: 'types of triangle'; 'triangle inequalities', 'triangle trigonometry'. Assuming no information about the user is available, covering these aspects in the displayed result is necessary (so that a user can both discover about their existence and have a comprehensive view on the topic, as argued by [27]), but not sufficient. For a better learning experience that would best satisfy many users' learning needs, we need to diversify the result set with respect to *each* of possible dimension (e.g. topicality, level of difficulty, type of documents, etc.).

The diversification of results taking several dimensions into account is both a related and a more complicated research problem than the flat (topical) and hierarchical diversification, which have been widely investigated for general-purpose web search. While the latter approaches only need to discover the topical aspects of a query, multi-dimensional diversification needs the dimensions to be related to a query, as well as the aspects related to each dimension. We envision that, for general-purpose search engines, it may not be necessarily optimal to consider diversification over all the aforementioned dimensions for every query[2] since (i) it will interfere with many other signals for ranking and cause quality reduction for non-educational queries, and (ii) the knowledge of all the dimensions and their semantics may not be readily available. Therefore, different from most existing works, rather than a general-purpose search system, our work focuses on an educational search engine, where both the collection (i.e., educational materials) and users of the system would have a richer set of features that could be naturally exploited for search towards learning. Consequently, we examine the multi-dimensional diversification of search results in this context. We employ the data from a real-life educational search engine embedded into a commercial web-based educational framework for K-12 level students in Turkey with around 1.2M registered users.

[2] As will be discussed in the Related Work, there are several earlier works diversifying the results only for ambiguous queries when a single (topical) dimension is used.

In this chapter, we investigate the impact of the newly proposed multi-dimensional result diversification approach in the context of educational search. Our contributions are as follows:

1. We define a new result diversification problem that addresses the typical requirements of a search as learning scenario, i.e., where there are a wide range of dimensions the search engine needs to consider when returning results that meet the learning goals (i.e., providing comprehensive information on the topic in many forms (e.g. various types of documents) and at many education levels (e.g. from level 4 to 8)).

2. We provide a new framework for diversification, which extends the state-of-the-art diversification methods (namely, xQuAD [15]; a variant of PM2 [16] as well as a supervised approach, R-LTR [17]), to handle multiple dimensions, and provide tailored instantiations for the framework. Specifically, we enrich each diversification method so that while an aspect's coverage in the final ranking is computed, the importance of the dimension which this aspect belongs to is also taken into account. To illustrate our motivation for computing dimension importance values, let an example query be 'triangle', and assume that the candidate set has documents from all the types available in the system but they all pertain to the education level 4. In this case, the diversification algorithm should focus on diversifying documents w.r.t. the type dimension, since there are several aspects to cover there, but should not attempt to diversify for the education level dimension. Hence, while setting the importance values for certain dimensions adaptively (i.e., per query), we consider the variety of the aspect values observed in the candidate set.

3. We describe a new rich dataset[3] tailored for the evaluation of diversification algorithms with multiple dimensions, built from user interactions with an existing real-life educational search engine.

4. We carry out an extensive evaluation of our work using a realistic experimental set-up, which is based on query instances and clicks in addition to TREC-style relevance annotations. Our experiments demonstrate the effectiveness of our

[3] `https://github.com/syigitsert/multi-dim-diversification`

proposed approach in comparison to strong baselines, showing improvements of 2.6%, 1.4%, and 2.2% for the diversification metrics ERR-IA, α-nDCG and P-IA, respectively; and an improvement of 1.4% for the traditional P@2 metric. Considering the positive impact of diversified result presentation on the learning outcomes (e.g., knowledge gains of users) as shown in [28, 29], these improvements in diversification performance are likely to translate into learning gains in the educational search context, which is the ultimate goal of our present investigation.

2.2 Related Work

2.2.1 Search as Learning in General-Purpose and Educational Search Engines

Exploiting search for learning is a recent research direction attacked from various perspectives [10, 23, 24]. Some of the previous works aim to shed light on the impact of search activity on a user's knowledge or expertise (e.g., [30, 31]). Others attempt to enhance the performance of search engines for core tasks, most crucially, the ranking of documents, to support learning-oriented search (e.g., [27, 28, 32]). Note that, while the latter works address general-purpose search engines, where only a fraction of queries might be towards a learning goal, there is also an emerging line of research on the *"semi-informal learning settings that involve search for scholarly information"* [24] and *"educational information systems"* that may be utilized by teachers and/or students [10].

We see the latter setup, broadly referred to as educational search engines here, as a complementary direction with new opportunities and challenges, and propose our multi-dimensional diversification framework in this context. In the following, we briefly review the earlier works in the aforementioned areas.

Enhanced ranking in general-purpose and educational search engines for learning goals. A particular direction to enhance learning experience via search involves the re-ranking of results from general search engines. In one of the pioneering studies, Collins-Thompson et al. [25] proposed to personalize Web search results by re-

ranking them with respect to reading difficulty.

Azpiazu et al. [26] analysed childrens' and teachers' needs while searching, and introduced a search framework, YouUnderstood.Me (YUM), based on the results of an existing search engine. YUM returns documents that are compatible with the reading abilities of users while satisfying their information needs. More recently, Yilmaz et al. [32] proposed an approach in which they trained classifiers using various educational resources to predict the related course category of question-like queries, and then employed these predictions as a signal for re-ranking the initial query results. In contrary to all these studies enhancing the results from a general-purpose search engine, our work employs a more specific – i.e., educational – search setup. Furthermore, we leverage diversification as the key methodology to obtain a re-ranking of results, so that all users with diverse topical interests and diverse proficiencies/preferences may benefit from the search process; in contrast, the aforementioned works employ a different re-ranking methodology, which is based on customizing the final result list w.r.t. a single feature of a given user (i.e., reading level) or query (i.e., related course category).

To the best of our knowledge, there are only two other studies that employ diversification in a learning-related context. In [27], from a broader perspective, the authors addressed the exploratory Web search queries and so-called *intrinsically diverse* sessions, where users aim to learn about a topic by seeking information about its multiple aspects. To address such queries, they introduced a greedy diversification algorithm that re-ranks the initially retrieved results. Syed and Collins-Thompson [28] presented a retrieval algorithm to enhance the educational benefit in the vocabulary learning task. In this task, users learn essential keywords for a particular topic by reading documents covering different aspects of the topic. Again, over an initial set of results obtained from a general-purpose search engine, they apply a diversification algorithm (based on the intrinsic diversity algorithm of [27]) that also takes into account the keyword density, as the goal is covering a diverse vocabulary. Our work is similar to those as we also build on diversification, however, we employ a more specific educational search setup that enables us to apply diversification across multiple dimensions, but not only for a single dimension (i.e., topical aspects) as in the latter works.

In contrast to the aforementioned works that aim to improve the general purpose search engines to support search for learning, an alternative and complementary research direction is focusing on specialized educational/learning settings that also involve search. For instance, Hoppe et al. [24] mentioned TIB's web portal[4] as one such semi-formal setting, where one can search for scientific videos. A commercial example is iSEEK Education[5], a targeted search engine for learners based on the resources from universities, governments, etc. Usta et al. [33] presented an analysis of an educational search engine that works on a proprietary education platform in Turkey for K-12 students with 1.2M registered users. In the same setting, Vidinli and Ozcan [34] proposed a query suggestion approach. In this work, we also focus on an educational search engine setup, as it serves as a natural testbed for the proposed diversification approach across multiple dimensions.

Evaluating the impact of search on learning. A particular strategy to evaluate the impact of a search session on users' knowledge gain on a certain topic or domain is by conducting pre- and post-assessments via tests, summaries, or user studies. For instance, Collins-Thompson et al. [29] focused on the impact of various of Web search strategies (i.e., results of the single- and multi-query searches as well as those yielded by their intrinsic-diversification approach) and evaluated them via questionnaires and self-reports. Moraes et al. [31] compared the users' knowledge gain on a topic in various scenarios including search session(s), versus the baseline scenario of watching an instructor-designed video. They evaluate each user's knowledge on a given topic using the vocabulary knowledge test (applied to participants before and after each scenario), and show that watching the video together with search led to the highest gain in knowledge. Due to the requirement of some form of explicit feedback from the users, such an evaluation strategy is more applicable in lab settings and with a limited number of participants. In our work, as we exploit real search logs from an educational search engine but do not have a chance to interact with the actual users, we cannot apply such direct assessments. Thus, we rely on traditional metrics computed over the re-ranked results using the proposed multi-dimensional diversification framework. Having said that, the aforementioned work of Collins-Thompson et al. [29] has already shown that an intrinsically diverse presentation of search results yields the

[4] `https://av.tib.eu`
[5] `http://education.iseek.com/iseek/home.page`

highest percentage of users' with knowledge gains; and hence, our improvements in terms of the traditional and click-based diversification metrics imply a high likelihood of improving human learning in an educational search context. Several earlier works also employed proxy signals (such as dwell time and user clicks) to assess the user learning via search. For instance, Eickhoff et al. [30] conducted a query log analysis to detect expertise development within and cross search sessions. In other works that are more similar to ours, i.e., proposing to re-rank search results taking into account the readability level [25] or intrinsic diversity [27], the authors again relied on traditional metrics, as we do in this study.

2.2.2 Diversification of Search Results

Result diversification methods for ambiguous queries. In the literature, diversification approaches are essentially applied to ambiguous queries (such as the query 'jaguar', which could be seeking information for either the aspect 'animal' or 'car') where the user's search intent cannot be clearly determined.

Santos et al. [6] suggested that methods of diversification can characterized as either *implicit* or *explicit*, which differ in how the diversification is conducted. In particular, implicit approaches only inspect attributes of each document itself, usually their contents. For instance, Maximum Marginal Relevance (MMR) [35] is an early (implicit) diversification method that aims to balance between the relevance of documents to the user's query, and the diversity in the contents of those documents. Other implicit diversification approaches employ methods based on Probabilistic MMR [36], risk minimization [37], portfolio theory [38], matrix factorization [39] and clustering [40, 41]. Zhu et al. [17] proposed a supervised implicit diversification approach, R-LTR, that learns the weights of its scoring function that reminds MMR, employing Stochastic Gradient Descent (SGD).

In contrast, explicit approaches use an external representation of the possible information needs of the user (also known as *aspects*), and aim to ensure that many of those aspects are covered in the top-ranked documents. For instance, in the IA-Select method [4], a user query is positioned within the categories of the Open Directory

Project[6], and as many categories as possible are surfaced in the top-ranked results. Similarly, the xQuAD diversification framework [15] recommends the use of common query reformulations (as mined from the log of the previous user queries). Such reformulations show the varying intents of users underlying an ambiguous query that should be surfaced by diversification within the results for that query. Other approaches to diversification, such as PM2 [42] performs diversification that is relative to the popularity of explicit representations of the possible aspects of the query. Jiang et al. [9] apply supervised learning methods, and specifically neural networks with the attention mechanism, to model the coverage of query aspects by the documents and obtain diversity scores.

In general, explicit approaches to diversification have been shown to be more effective in evaluations such as the TREC Web track, as they better represent the possible information needs underlying an ambiguous query [6]. Our work takes a hybrid viewpoint to diversification, by diversifying by both explicit representations of the *topical* ambiguity, as well as implicit aspects of the documents, such as the way they present their information (e.g. tutorial, video) or their readability by different age groups of user, which may imply their suitability for different users. In this way, our work considers multiple *dimensions* that naturally arise in an educational search setting for diversification.

There are three existing works from the literature are closer to ours in terms of the diversification methodology. Firstly, Hu et al. [43] introduced the notion of hierarchical intents of topicality. Our work goes further by considering multiple orthogonal dimensions of diversification rather than a strict hierarchy, and goes beyond topicality, to encompass other dimensions that can be estimated (e.g. readability) or derived from document metadata attributes (e.g. document type). Secondly, Aktolga [16][Ch.5] investigated adaptations to PM2 that could achieve a mixed diversification of both topical and non-topical (implicit) dimensions, namely, the sentiments and dates expressed in the documents. Finally, Dou et al. [44] proposed a multi-dimensional topic richness model in a similar fashion to xQuAD for web search diversification. They considered each dimension as a *data source* (such as anchor texts, query logs, web sites, etc.) from which different and mostly *topical* aspects can be mined. Hu et al. [45] ex-

[6] `http://dmoz.org`

tended the latter approach with the aspects derived from an additional data source, namely, the lists appearing in the candidate documents. In contrast to the latter approaches, our experiments focus on dimensions of diversification that are appropriate to an educational search engine. Furthermore, we also extend R-LTR, an implicit diversification method, to exploit explicit aspects for multiple dimensions. To the best of our knowledge, R-LTR has been used with explicit aspects only in [46], but again, not for handling dimensions in the context of educational search. Last but not the least, none of these approaches employ a click-based evaluation setup as we do in this study.

Result diversification methods for learning. As discussed in the previous section, there are a few works that apply diversification specifically towards learning-related tasks. Raman et al. [27] introduce a diversification method to re-rank search results for the so-called *intrinsically diverse* search sessions, where users aim to explore and learn about a particular topic. The distinction of *intrinsic* and *extrinsic* diversity is based on the following observation [27, 47]: Intrinsically diverse sessions are those where users submit several related queries for the different aspects of the *same* topic (say, for the query 'jaguar', aspects are like 'jaguar feeding habits', 'jaguar habitat', etc.). In contrast, extrinsic diversity arises due to ambiguous or underspecified queries (e.g., for the query 'jaguar', aspects may include 'jaguar animal' and 'jaguar car'). Indeed, extrinsic diversity has motivated most of the diversification methods described in this section. The methodology presented by [27] for the mining of intrinsic queries is not completely different to that employed by [48] for detecting ambiguous queries, as both involve identifying the related queries of the main (initiator) query appearing in the current session, although the latter also considers multiple sessions for a given main query (and hence, obtains broader aspects). In our work, we believe that the distinction between extrinsic and intrinsic diversity is less visible, as the queries submitted to an educational search engine, by definition, share a narrower scope, e.g., are concerned with the subjects related to various courses. Hence, while creating the dataset for this work, we followed the procedure of [48] for identifying the main query set to be diversified. At the end of this process (described in Section 2.4.1), we found that the majority of the queries exhibit similarity to intrinsic queries of [27], i.e., they seek information for the various aspects of the same main topic (e.g., for the

query ‘triangle’; aspects are like ‘types of triangle’; ‘triangle inequalities’, ‘triangle trigonometry’), while just a few of them involved some ambiguity (e.g., the query ‘voice’ (in Turkish) has aspects related to both physics and language). Finally note that, the work of Raman et al. [27] evaluates performance using typical relevance metrics (considering satisfied (SAT) clicks as relevant), while we exploit both traditional annotated judgments and clicks. Another practical difference of the latter method in [27] is that the candidate result set to be diversified is formed by the union of top-ranked documents retrieved for the main query and its aspects, while most of the other diversification methods operate on a candidate set retrieved only for the main query. As discussed above, in our educational search setup, where the underlying collection is not as large and diverse as Web, the documents retrieved for a query and its various aspects would not hugely differ (but their ranking would); and we proceed with the typical setting, where diversification is applied over the candidate set retrieved for the main query. In [28], a modified version of the diversification algorithm of [27] is employed for a vocabulary learning task. For this latter task, the authors conducted a crowdsourced user study and directly evaluated the attained learning outcome. As we discussed in depth in Section 2.2.1, such an evaluation is not possible in our setup; yet based on the findings of [28] and [29] revealing the positive impact of diversified result presentation on the actual learning outcomes, we believe that using traditional metrics in our evaluations is reasonable.

Other applications. While our focus is on search result diversification in this chapter, diversification has also been investigated for recommendation systems, in order to make recommendations more useful. Vargas et al. [49] proposed a Binomial framework that provides diverse recommendations based on genre information of movies using coverage, the redundancy of genres and also regarding the size of the recommendation list. Their study aims to enhance recommendations by taking advantage of diversification. Unlike our study, they only considered one dimension, namely genre, while diversifying. Di Noia et al. [50] assess the propensity of each user towards diversity before personalization of the recommendations. Diversity is applied by taking multiple attributes (i.e. genre, year, actor etc.) of items into account. By incorporating user propensity, the diversification algorithms (xQuAD and MMR) re-rank the top-N recommendation list that is computed per person by a recommendation algorithm.

Overall, these studies relate to recommendation systems while, conversely, we aim to improve search experience. Moreover we also focus on a different domain, namely that of education.

2.3 Diversification across Dimensions

In this section, we introduce the xQuAD [15], PM2 [42] and R-LTR [17] diversification approaches, and show how to adapt both to consider multiple dimensions.

2.3.1 xQuAD

xQuAD iteratively selects documents from an initial ranking of candidate documents for query Q, denoted $R(Q)$, into the final ranking S that maximizes the following objective:

$$(1-\lambda)\Pr(d|Q) + \lambda \sum_{a \in Q} \Big[\Pr(a|q) \Pr(d|a) \prod_{d_j \in S} (1 - \Pr(d_j|a)) \Big], \tag{2.1}$$

where Q is the user's query, a is an aspect of Q, and S is the set of already selected documents. $\Pr(d|Q)$ and $\Pr(d|a)$ are identically defined, in being the score of a document with respect to the original query, or an aspect, and can be calculated using any effective document ranking approach, such as BM25 [15] or more advanced learned ranking models (e.g., [6]). $\Pr(a|q)$ represents the importance of that aspect for the query, and, by default, is uniform across all aspects [15][7].

We note that the novelty $\prod()$ component of xQuAD, can be referred as $\Pr(\bar{S}|q_i)$, may yield small values as more documents are selected into S and the corresponding $(1 - \Pr(d_j|a))$ values are multiplied [22]. As a remedy, the product can be replaced by with the geometric and arithmetic mean of the probabilities [22], as shown in Equations 2.2 and 2.3, respectively. We refer to these variants as art_xQuAD and geo_xQuAD hereafter.

$$\Pr(\bar{S}|q_i) = \frac{\sum_{d_j \in S} (1 - \Pr(d_j|q_i))}{|S|} \tag{2.2}$$

[7] In Section 2.5.2, we go beyond this assumption and learn the aspect importances from the user clicks.

$$\Pr(\bar{S}|q_i) = \sqrt[|S|]{\prod_{d_j \in S} (1 - \Pr(d_j|q_i))} \tag{2.3}$$

Overall, xQuAD can diversify across any intent space Q, but, typically, common query reformulations are used to identify topics the user may be looking for. As discussed above, this omits the possibility that there might be other independent factors (i.e., dimensions) affecting a document's suitability for the users.

2.3.2 Multi-dimensional xQuAD

We assume that there are multiple dimensions of diversification $dim \in D$, possibly conditioned on the query Q (denoted $D(Q)$), which we wish to be covered in a ranking. Each dimension dim has a corresponding set of aspects: $a_1, ..a_n$. For the topic dimension, which is generally applied in web search, aspects are the underlying intents that can be inferred via mining the query reformulations or knowledge-bases. Although our model is more general, in this study we consider two further dimensions, namely the (educational) level that the document targets and the type of document, which are specific to our target application of search in the education domain. The aspects for such dimensions may also be identified in similar ways to the topic dimension, e.g., for a given query, we can utilize related suggestions and their retrieved (and even clicked) results to detect the relevant educational levels or document types.

Our proposed model is straightforward in that it adapts xQuAD by marginalising over all dimensions, as follows:

$$(1 - \lambda) \Pr(d|Q) + \lambda \sum_{dim \in D(Q)} \sum_{a \in dim} \Big[\tag{2.4}$$
$$\Pr(dim|Q) \cdot \Pr(a|dim, Q) \cdot \Pr(d|a, dim) \cdot \prod_{d_j \in S} (1 - \Pr(d_j|a, dim)) \Big].$$

In Equation 2.4, the component probabilities are: $\Pr(dim|Q)$ defines the *dimension importance*, which represents the importance of dimension for the query; $\Pr(a|dim, Q)$ defines the *aspect importance*; and $\Pr(d|a, dim)$ is the *document aspect coverage*. Note that we differentiate between the dimensions for which the probability $\Pr(d|a, dim)$ should be *estimated* (such as the relevance of a document to a topical aspect of a

Table 2.1: Dimensions & aspects used in this work.

Dimension	**Aspects**	$\Pr(d\|a, dim)$ **value**
Topicality	via log mining	Estimated
Education Level	$\{4, 5, 6, 7, 8\}$	Known
Type	{animation, interactive exercise, video, text, game, lecture, conversational exercise, application, summary}	Known

query) and dimensions for which this probability can be accurately known based on the available metadata for documents (e.g., given a query related to the *animation* aspect for the type dimension, the diversification algorithm can assign the probability, $\Pr(d|a_{animation}, dim_{type})$, to either 0 or 1 based on the metadata associated with the document d). Table 2.1 highlights the dimensions and aspects that we consider in this study. Note that, as mentioned in the introduction, while the aspects for the education level dimension typically cover the range of 1 to 12, for K-12, our query log sample covers only the range of 4 to 8.

To instantiate the proposed multi-dimensional xQuAD approach, we discuss how to instantiate the dimension and aspect importance probabilities. In particular, the importance of diversification upon each dimension may vary between queries – for example, observing documents with a variety of education levels in the candidate set of documents $R(Q)$ for a particular query Q may suggest that portraying these different levels of content (c.f. Table 2.1) in the top-ranked documents are likely to benefit a wide range of users. Thus, for the (education) level dimension, we set the dimension importance as follows:

$$\Pr(dim_{level}|Q) = \frac{O(Q) - min_{level}}{max_{level} - min_{level}}, \tag{2.5}$$

where $O(Q)$ denotes the level aspects observed in $R(Q)$, and max_{level} and min_{level} denote the maximum and minimum number of possible aspects in the level dimension. For instance, if the documents in $R(Q)$ cover the level aspects {5, 6, 7} and all possible level aspects are {4, 5, 6, 7, 8}[8], we set $\Pr(dim_{level}|Q) = (3-1)/(5-1) = 0.5$. Note that, if all possible aspects are observed in $R(Q)$, the importance score is 1, while if only one aspect is observed, it is 0; i.e., no need to diversify for this dimension. The importance of the type dimension is set in the same manner. How-

[8] This is because our dataset covers the range [4, 8] for the education levels.

ever, for the topic dimension, we cannot know how many aspects are observed in the candidate set, as we can only *estimate* topical relevance. Hence, we intuitively set $\Pr(dim_{topicality}|Q) = 1$, as we expect relevance to be the first driver of diversification, with diversification encapsulating other dimensions having relatively lesser importance.

2.3.3 PM2

PM2 [42] adapts the allocation problem of seats to party representatives in some election systems to finding a diversified result list. The diverse result set is constructed with respect to the set of aspects related to the query in proportion to the popularity of these aspects. PM2 starts with a ranked list, $R(Q)$ which represent the candidate documents, with k empty seats which is the size of the diversified list, S. In each iteration, the winner aspect is determined by the popularity of the aspect (referred as quotient score). The quotient score is computed for each aspect i via:

$$quotient[i] = \frac{v_i}{(2s_i + 1)} \tag{2.6}$$

where v_i and s_i indicate the number of votes the party i receives and the number of seats that have been assigned to the party i. A seat (the position in S) is allocated for the winner aspect, i.e. i^*, and the document d^* that is relevant to the winner aspect is selected by the following score function:

$$d^* \leftarrow \underset{d_j \in R(q)}{\arg\max}\ \lambda \times qt[i^*] \times Pr(d_j|q_{i^*}) + (1-\lambda) \sum_{i \neq i^*} qt[i] \times Pr(d|q_i) \tag{2.7}$$

where $qt[i]$ is the quotient score and λ is the trade-off parameter between the relevance to the winner aspect and other aspects. Since the selected document is relevant the other aspects in addition to the winner aspect, PM2 updates the portion of seats in the selected set, S.

2.3.4 Multi-dimensional PM2

We adapt the original PM2 for diversification with multiple dimensions following a similar approach to that of [16][Ch.5]. Unlike the original PM2 formulation, we have

one s_i, which indicates the portion of selected documents in S for aspect i, and v_i, which denotes the number of documents the aspect i should have, for each aspect a of each dimension $dim \in D$. The quotient score is calculated for each aspect a under each dimension dim. Multi-dimensional PM2 selects the winner aspect i^* for each dimension, and then computes relevance of the next document for S to the winner aspect versus its relevance to all the other query aspects within that dimension.

Note that our approach is similar to the adaptation of PM2 to multiple dimensions proposed by [16][Ch.5], in the choosing of a winning aspect i^* for each dimension. However, it differs in terms of computing dimension importance and λ parameter. We use Equation 2.5 for dimension importance instead of interpolation weights and λ parameter without smoothing. Besides non-topical dimensions differ, in our approach, topical dimension is incorporated explicitly instead of classifying documents with respect to their sentiments. Sentiment classification may not be reasonable for every domain, especially where the majority of documents have neutral sentiment (i.e., as in our case, education).

The scoring equation of multi-dimensional PM2 is as follows:

$$d^* \leftarrow \underset{d_j \in R(q)}{\arg\max} \sum_{dim \in D(Q)} Pr(dim|Q) \times \lambda \times qt[i^*, dim] \times Pr(d_j|q_{i^*,dim}) + (1-\lambda) \sum_{i \neq i^*} qt[i, dim] \times Pr(d|q_i, dim) \quad (2.8)$$

We use the same setting to instantiate the dimension and aspect probabilities for multi-dimensional PM2 as in multi-dimensional xQuAD.

2.3.5 R-LTR

R-LTR [17] is a supervised implicit diversification method that learns the weights of its scoring function using SGD. Given a candidate document set $R(Q)$ for a query Q, R-LTR constructs the final ranking S in a greedy manner. In each iteration, R-LTR computes the following scoring function for each document d_i that is not in the ranking S, and the one with the highest score is added to S:

$$\text{R-LTR}_{\text{imp}}(d_i, V_i, S) = \omega_r * x_i + \omega_d * h_S(V_i) \quad (2.9)$$

The first part of Equation 2.9 represents the relevance of the scored document, and the second part represents its diversity from the documents already selected in the ranking S. x_i denotes a relevance feature vector that comprises scores expressing query-document matching (e.g., tf-idf, BM25, etc.), while V_i is a matrix capturing the diversity scores of d_i to all other documents in $R(Q)$, in terms of various diversity functions. Hence, V is a 3-way tensor that stores the diversity between each pair of documents in $R(Q)$, each computed using various diversity functions. Finally, ω_r and ω_d are the weight vectors (for the relevance and diversity components, respectively) that are learnt during the training stage.

Since the original R-LTR is an implicit method, it does not employ the knowledge of aspects. Thus, in our setting, for a given pair of documents, the diversity scores (stored in the tensor V) are computed as follows. First, based on the content of the documents, we compute two different diversity scores using typical similarity measures from the literature: i) the tf-idf weighted Cosine similarity, and ii) the Jaccard Coefficient of the document vectors. Second, since the candidate documents' education level and type information are also available (as metadata) during the diversification, we compute their distance using Jaccard Coefficient and Binary Similarity Coefficient, respectively. Thus, for each pair of documents, the tensor V stores a vector of 4 different diversity scores. Note that, in Equation 2.9, while computing the diversity of d_i to the documents already selected in S, the aggregation function $h_S()$ is invoked, which is the minimum function in our setting (as in [17]). We denote this baseline by R-LTR$_{\text{imp}}$.

2.3.6 Multi-dimensional R-LTR

We propose a variant of R-LTR that uses the explicit aspects associated with multiple dimensions, as described in the previous sections. In Equation 2.10, V^{topic}, V^{level} and V^{type} store the pairwise diversity scores (utilizing the associated aspects) across topic, education level and type dimensions, respectively.

$$\text{R-LTR}_{\text{exp}}(d_i, V^{topic}, V^{level}, V^{type}, S, Q) = \omega_r * x_i + \Pr(dim_{topic}|Q) * \omega_{topic} * h_S(V_i^{topic}) + \Pr(dim_{level}|Q) * \omega_{level} * h_S(V_i^{level}) + \Pr(dim_{type}|Q) * \omega_{type} * h_S(V_i^{type}) \quad (2.10)$$

In this case, for each dimension, documents are represented w.r.t. their relationship with the related aspects. Specifically, for the topicality dimension, we represent each document as a vector of $\Pr(d|a_i)$ scores, which is the score of the document d with respect to each aspect a_i (calculated using an effective ranking approach such as BM25). Then, the tensor V^{topic} stores the Euclidean distance between these document vectors, as a particular type of diversity score. Furthermore, we calculate the pairwise difference of the $\Pr(d|a)$ scores between two document vectors and obtain their maximum and minimum as additional diversity scores. For the level and type dimensions, as before, we assign binary values for each aspect according to the metadata associated with the document d. We compute the Euclidean distance between the document vectors to be used as diversity scores in the V^{level} and V^{type} tensors. Finally, for each dimension, the aggregated score is multiplied with $\Pr(dim_i|Q)$, which is instantiated as before. We call this method R-LTR$_{\text{exp}}$.

Finally, in [51], R-LTR was implemented using a neural network framework, which allows a non-linear formulation and the training of more complex models (i.e., via multiple hidden layers). Similarly, we apply this approach for training a model based on the same input as Equation 2.10 (denoted by R-LTR$_{\text{expNN}}$).

2.4 Dataset

We now describe a new dataset that we have constructed in the context of an educational search engine as a benchmark to evaluate the existing and proposed methods for the multi-dimensional diversification problem. We cannot use an existing TREC dataset as (1) they do not have dimension information and corresponding relevance judgements, and (2) there is no specific dataset for the educational search context that we aim to address in this work. Expanding upon the topic development practice of the TREC Web Track Diversity tasks [18], the dataset is created as follows.

2.4.1 Identifying the main queries

As our starting point, we use a query log from an educational search engine embedded into a commercial web-based educational framework (called Vitamin[9]) for K-12 level

[9] http://www.vitaminegitim.com/

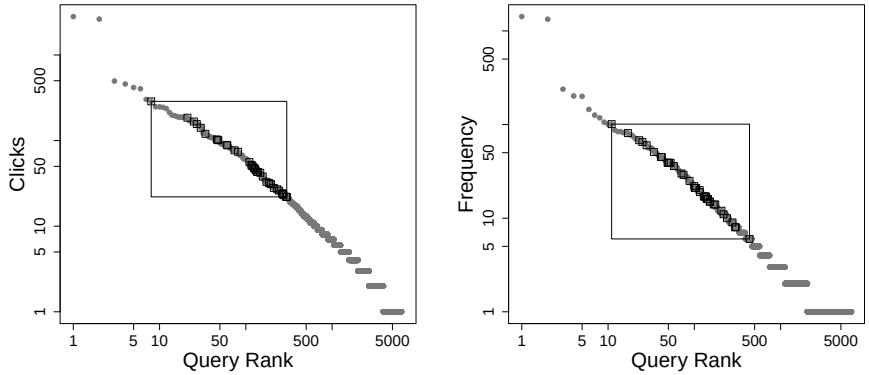


Figure 2.1: Distribution of query click count (left) and frequency (right); our main queries are sampled from the marked regions in each plot.

students[10] in Turkey with around 1.2M registered users. The query log contains a sample of 20K queries (6,503 of which are unique) from April 2015. To identify the main queries that would benefit from diversification, we follow [48] and use click entropy which indicates the variation of the clicked documents for each query. The selected queries have a total click count of 20 or more, and an entropy greater than 1.5. We also manually eliminated the near-duplicate queries that are extremely short or textual variations of each other (i.e., 'triangle' vs. 'triangles'), keeping the variant with higher entropy. For the remaining queries, we obtained their related queries (i.e., a related query to q is a query $q\prime$ either following q in a search session, as in [18], or completely including the query string q). We then discarded the queries with no or trivial related queries or with completely non-relevant ones (i.e., no relevance to the original query). This procedure yielded us 40 queries, such as 'light', 'angles', 'electricity' that have a variety of aspects.

With respect to the total click count and occurrence frequency, these 40 queries come from the "torso" of the power law distribution of the query log (as shown in Figure 2.1), and hence, seem to be representative of the query volume (as in [18]). Head queries, such as 'mathematics', 'game', 'science', etc., are too generic and it is unreasonable to determine a set of possible aspects underlying those queries. For tail queries (e.g., 'converting poetry to prose'), there is nothing to diversify since they are

[10] i.e. targeting students aged 5-17 and covering primary, middle and high school educations (as in U.S.A.).

very specific in nature. Note that the majority of our main queries exhibit similarity to intrinsic queries of [27], i.e., seek information for the various aspects of the same main topic (e.g., for the query 'triangle'; aspects are like 'types of triangle', 'triangle inequalities', 'triangle trigonometry'), while just a few of them exhibit extrinsic diversity and involve ambiguity (e.g., the query 'voice' (in Turkish) has aspects related to both physics and language).

2.4.2 Identifying ground-truth aspects for the topic dimension

Firstly, we clustered the reformulations of a query (using a hierarchical clustering algorithm, as in [18]) to determine candidate aspects. Next, we applied a manual post-processing process for the noisy clusters, i.e., we merged the clusters that are clearly related to the same underlying aspect, and removed those clusters that are redundant, i.e., including queries related to aspects already represented by other clusters. Finally, human annotators evaluated these aspects taking into account the domain knowledge (i.e., even if there does not exist a cluster labelled as "trigonometry" for the "triangle" query, the annotator may add it as an aspect). While doing so, the annotators took into account the retrieved documents as listed in the query log, as well as the knowledge obtained from other educational resources. Note that these aspects serve as the "official" ones for the topic dimension (we discuss the aspects for the type and level dimensions later, see Section 2.4.4).

2.4.3 Automatically extracting aspects for the topic dimension

In the literature, explicit diversification algorithms are usually evaluated using both *official* aspects (i.e., representing an ideal scenario where aspects are known perfectly) and those that are automatically extracted by various intent mining approaches. In our benchmark dataset, we also provide a set of aspects (for the topic dimension) that are created without any manual intervention. In particular, for each query, we simply clustered its reformulations and obtained the cluster label (i.e., the most frequent 3 terms in the cluster [52]), which serves as the extracted aspect. Furthermore, we used a temporally disjoint query log (i.e., from Dec. 2013) to automatically create another

set of aspects using the same procedure. These two sets of aspects are referred to as Auto2015 and Auto2013, respectively.

2.4.4 Document-level annotation

For each main query in our set, we first obtained all of its occurrences in the query log, and constructed a union of the results (namely, top-25 documents) for each occurrence. Note that, these top-25 documents may slightly differ for different occurrences of the query, due to updates in the collection, etc. Then, for each query, we created a matrix-like evaluation form where the rows are the documents (with titles and short descriptions) and the columns are the main query and its aspects in the topic dimension. We do not need to manually annotate the documents with respect to the aspects of the level and type dimensions, as this information can be inferred from the documents' metadata (see Section 2.3.2). We used 5 judges (all with a computer science background) to annotate the (binary) relevance of each document to each aspect of the queries.

In general, the documents for each query were annotated by one judge, yielding a total of 12,735 annotations. For a random subset of 4,842 annotations, we also employed a second judge. The Cohen's κ coefficient of inter-rater agreement on these 4,842 annotations is 0.77, which indicates substantial agreement [53]. The observed level of agreement suggests that the relevance annotation task is fairly easy for the used query set, and hence, our choice of assigning a single judge per query is adequate.

Finally, for the type and level dimensions, we obtained the official aspects of a query by accessing the metadata of the topically relevant documents in the ground-truth for that query. On average, this yielded 3.55 and 4.53 aspects per query for the type and level dimensions, respectively.

2.5 Experimental Evaluations

We essentially consider 2 different frameworks for the evaluation of our proposed methods. The first framework is based on the relevance judgments, i.e., annotations, that are obtained via TREC-style topic development and user evaluation procedure,

as described in the previous section. In this case, the evaluation *metrics* (such as P-IA, etc.) are computed over the diversified result list of a query under the traditional assumption that all query aspects are equally likely. (Indeed, probably due to the latter fact, most works in the literature obtain the best diversification results when the aspect importance component, i.e., $\Pr(a|q)$ in Equation (2.1), is uniformly distributed across all aspects, as we also assume in Section 2.5.1.).

However, in practice, query aspects may have varying importance (or, popularity) from the *user perspective*; e.g., for the query “light", we observed that much more users (students) click on the results related to the “absorption of light" aspect than those related to the “light year" aspect. Hence, to evaluate the diversification performance by taking actual user preferences into account, we devise a second framework where evaluation metrics are essentially computed using the clicked results for each query instance, separately. As we discuss in Section 2.5.2, such a framework also gives a way to set the aspect importance probabilities more realistically (i.e., by learning from a training set), as it can truly evaluate the differences in these priors.

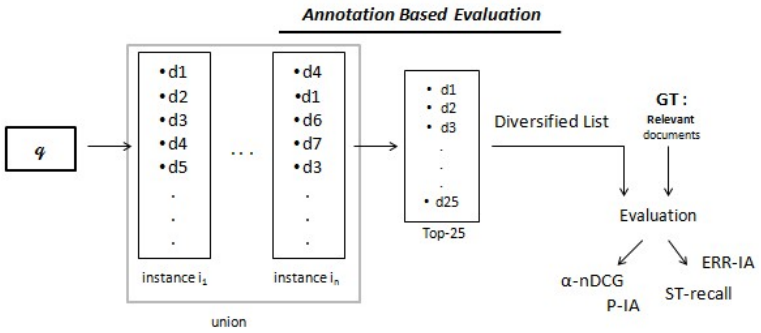
We refer to these evaluations as annotation- and click-based frameworks, and provide detailed set-ups and experimental results in Sections 2.5.1 and 2.5.2, respectively. Figure 2.2(a) & (b) visualize these frameworks and highlight their differences.

2.5.1 Annotation-based Evaluation

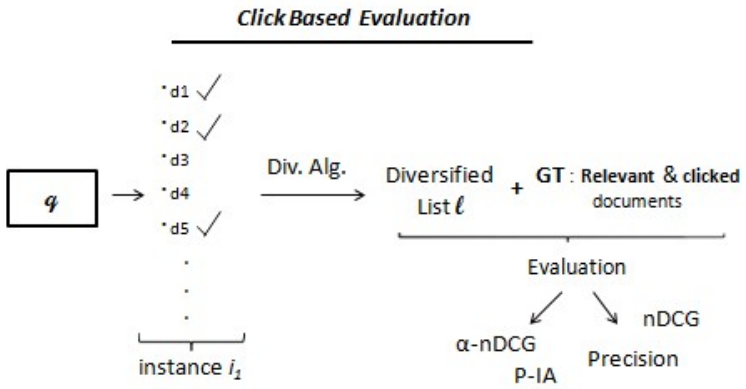
2.5.1.1 Setup

Initial retrieval and candidate set For each query, we re-ranked its result documents (i.e., the union of the results of all of its occurrences as obtained from the query log, as described above) using BM25 and identified the top-25 documents as the candidate set that is to be diversified.

Diversification methods & parameters As baselines, we essentially use xQuAD and the 2 variants with the novelty components employing the arithmetic (art_xQuAD) and geometric mean (geo_xQuAD) of the probabilities. We evaluate our multi-dimen-



(a)



(b)

Figure 2.2: Visualization of evaluation frameworks: (a) annotation-based, (b) click-based.

sional approach integrated into all 3 variants. In some experiments, PM2 and R-LTR are also employed.

We have 3 different dimensions to consider in diversification; education level, type and topic as specified in Table 2.1. For the topic dimension, we compute the relevance of the candidate documents (actually, their titles and short descriptions) to the main query and its aspects in the topic dimension, i.e., $\Pr(d|Q)$ and $\Pr(d|a)$, based on the BM25 scores as in earlier works. As the topical aspects, we experiment with the "Official" ones (as an ideal scenario) as well as the automatically extracted aspect sets Auto2013 and Auto2015. For the (education) level and (document) type dimensions, we assume that the official aspects appropriate for each query are not available at the time of diversification, which may be the case in practice. Hence, we obtain the education level and type of the documents that are in the candidate set of the query, and diversify only based on this knowledge.

For each dimension, we set the aspect probability $\Pr(a|dim, Q)$ assuming a uniform distribution across the aspects [11] in this dimension, as in the literature (e.g., [15]). For all methods, we report the results for the best-performing value of the trade-off parameter λ, which is 1. Note that, earlier works (such as [22, 54]) also report a similar value of λ, and attribute this due to use of Official query aspects.

Finally, R-LTR, being a supervised method, requires training. For all R-LTR variants, we set the learning rate (for stochastic gradient descent) to 0.005 based on the training data. To implement R-LTR$_{\text{expNN}}$, we train a fully connected two-layer neural network with back-propagation (using the PyTorch framework). The hidden layer has 10 nodes with a sigmoid activation function, and the number of epochs is set to 50. The ground truth ranking is obtained by greedily selecting the document that maximizes the α-nDCG metric as in [17]. We apply a 5-fold cross-validation to evaluate the performance.

Evaluation methodologies and metrics The ground truth includes the relevance labels of documents for the union of the aspects (of all dimensions), for a given query. Based on this ground truth, we compute typical and well-known diversification metrics in the literature, namely, ERR-IA [19], α-nDCG [20], Precision-IA (P-IA) [4] and Subtopic(ST)-Recall [36] and D♯-nDCG [55], all at rank cutoff 10. For computing the metric scores, we employ two approaches. The *Flat* evaluation is the traditional setup that does not take the dimensions into account, while the *DimAware* evaluation computes a metric score for each of the three dimensions and then obtains their average as the overall performance (i.e., as the layer-aware metrics in [56]). We use the Student's paired t-test (at 95% confidence level) for analysing statistical significance.

2.5.1.2 Results

Our experiments in this section seek answer to the following research questions to evaluate the performance of multi-dimensional diversification in the context of educational search:

[11] For the type and education level dimensions, we only consider the aspects observed in the candidate set.

Table 2.2: Diversification performances of Single Dimension and Flat xQuAD (using the Flat and DimAware evaluation). The superscripts with (†), (∗), (‡) denote a statistically significant difference from xQuAD Topic, Level, Type at the 0.05 level, respectively.

		Flat Evaluation					DimAware Evaluation				
Div	Method	ERR-IA	α-nDCG	P-IA	ST-recall	D♯-nDCG	ERR-IA	α-nDCG	P-IA	ST-recall	D♯-nDCG
None	BM25	0.425	0.766	0.304	0.796	0.817	0.447	0.766	0.321	0.811	0.848
Single Dim	xQuAD Topic	**0.468**	0.831	**0.309**	0.857	0.864	0.482	0.813	**0.325**	0.858	0.883
	xQuAD Level	0.437	0.792	0.299	0.852	0.839	0.457	0.791	0.316	0.867	0.869
	xQuAD Type	0.433	0.783	0.300	0.827	0.828	0.455	0.786	0.317	0.841	0.858
Flat	xQuAD	**0.468***,‡	**0.845***,‡	0.299†	**0.923**†,*,‡	**0.880***,‡	**0.483***,‡	**0.833***,‡	0.315†	**0.929**†,*,‡	**0.901***,‡

- Does using three dimensions altogether yield a better diversification performance than using each of these dimensions on its own?
- Do multi-dimensional xQuAD, PM2 and R-LTR variants (c.f. Sections 2.3.2, 2.3.4 and 2.3.6) yield better diversification than their so-called *flat* counterparts, i.e., the original algorithms (c.f. Sections 2.3.1, 2.3.3 and 2.3.5), respectively) that utilize all the aspects belonging to all dimensions as a flat set of aspects?
- Do our findings change when we use automatically extracted aspects for the topic dimension, instead of the Official ones?

To answer the first question, Table 2.2 compares the diversification effectiveness of three cases (using Official aspects for the topic dimension): i) non-diversified BM25 baseline, ii) original xQuAD algorithm that utilizes the aspects of each dimension, namely, topic, education level and type, separately; and iii) original xQuAD algorithm using the union of aspects from all three dimensions as a *flat* input. Our findings reveal that, diversification using aspects from even one dimension is superior to non-diversified baseline for the majority of metrics, and among the three dimensions, diversification via the topic dimension yields the best performance for all metrics. Furthermore, using aspects from all three dimensions (as a flat diversification) yields considerably better results than using a single dimension for most of the metrics. In other words, diversification considering just one dimension (say, topic) is not likely to

Table 2.3: Diversification performances of the flat and multi-dimensional methods (with the Uniform and Adaptive Instantiations of the dimensions' importance) using the Flat evaluation. In parentheses, we report the percentage change w.r.t. the corresponding flat method. The superscript (*) denotes a statistically significant difference using the Student's paired t-test (at 95% confidence level) w.r.t. the corresponding flat method.

		Flat Evaluation				
Div.	Method	ERR-IA	α-nDCG	P-IA	ST-recall	D♯-nDCG
None	BM25	0.425	0.766	0.304	0.796	0.817
Flat	xQuAD	0.468	0.845	0.299	0.923	0.880
	art_xQuAD	**0.477**	**0.865**	0.313	0.910	0.894
	geo_xQuAD	0.472	0.849	0.300	**0.925**	0.883
	PM2	0.475	0.862	**0.315**	0.914	**0.896**
M-Dim (Uniform)	xQuAD	0.467(-0.3%)	0.843(-0.2%)	0.299	**0.923**	0.880(-0.1%)
	art_xQuAD	0.476(-0.1%)	**0.865**	0.313	0.916	0.893(-0.1%)
	geo_xQuAD	0.469(-0.6%)	0.846(-0.4%)	0.299(-0.3%)	**0.923**(-0.2%)	0.880(-0.3%)
	PM2	**0.479**(0.9%)	0.861(-0.1%)	**0.316**(0.4%)	0.912(-0.2%)	**0.897**(0.1%)
M-Dim (Adaptive)	xQuAD	0.481(2.7%)	0.859(1.7%)	0.302(1%)	0.931(0.8%)	0.890(1.1%)
	art_xQuAD	**0.489**(2.6%)	**0.877**(1.4%)	**0.320***(2.2%)	0.913(-0.4%)	0.903*(1%)
	geo_xQuAD	0.482(2.1%)	0.860(1.2%)	0.301(0.5%)	0.929(0.4%)	0.889(0.6%)
	PM2	0.477(0.5%)	0.865(0.4%)	0.317(0.7%)	**0.936***(2.4%)	**0.908***(0.5%)

yield results that are also sufficiently diverse for the other dimensions. Although there may be some correlations between the aspects of different dimensions (e.g., topic and education level), the algorithms should better use all the dimensions explicitly for the best performance, as we aim to do in this work.

Table 2.3 addresses our second research question, i.e., can the multi-dimensional algorithms that explicitly model the query dimensions along with their aspects outperform their flat versions. To begin with, Table 2.3 (using the Flat Evaluation) shows that both the flat and multi-dimensional diversification methods (based on xQuAD and PM2) usually provide a notable improvement over the non-diversified BM25

Table 2.4: Diversification performances of the flat and multi-dimensional methods (with the Uniform and Adaptive Instantiations of the dimensions' importance) using the DimAware evaluation. In parentheses, we report the percentage change w.r.t. the corresponding flat method. The superscript (*) denotes a statistically significant difference using the Student's paired t-test (at 95% confidence level) w.r.t. the corresponding flat method.

		DimAware Evaluation				
Div.	Method	ERR-IA	α-nDCG	P-IA	ST-recall	D♯-nDCG
None	BM25	0.447	0.766	0.321	0.811	0.848
Flat	xQuAD	0.483	0.833	0.315	0.929	0.901
	art_xQuAD	**0.493**	**0**.854	0.329	0.927	0.915
	geo_xQuAD	0.487	0.838	0.316	**0.931**	0.904
	PM2	**0.493**	**0.855**	**0.331**	0.925	**0.919**
M-Dim	xQuAD	0.483	0.832 (-0.1%)	0.315	**0.929**	0.900(-0.1%)
(Uniform)	art_xQuAD	0.494(0.3%)	**0.856**(0.2%)	0.329(0.1%)	0.928(0.2%)	0.916
	geo_xQuAD	0.485(-0.4%)	0.834(-0.4%)	0.315(-0.3%)	**0.929**(-0.2%)	0.901(-0.3%)
	PM2	**0.497**(0.9%)	0.853(-0.2%)	**0.333**(0.5%)	0.923(-0.1%)	**0.921**(0.1%)
M-Dim	xQuAD	0.497(2.9%)	0.848(1.8%)	0.318(1%)	0.936(0.7%)	0.911*(1%)
(Adaptive)	art_xQuAD	**0.507**(2.8%)	**0.866**(1.4%)	**0.337***(2.2%)	0.923(-0.4%)	0.925*(1.1%)
	geo_xQuAD	0.498(2.1%)	0.848(1.3%)	0.317(0.4%)	0.934(0.3%)	0.909(1.3%)
	PM2	0.495(0.4%)	0.856(0.1%)	0.333(0.6%)	**0.943***(2%)	**0.930***(1.1%)

baseline for all metrics. For instance, while BM25 yields an α-nDCG score of 0.766, the best performing flat and multi-dimensional method, namely, art_xQuAD, yields 0.865 and 0.877, respectively. We also observe that, among the xQuAD variants and PM2, art_xQuAD consistently outperforms the others both for the flat and multi-dimensional cases (cf. the bold scores for each case in Table 2.3 and Table 2.4) for the α-NDCG metric.

Next, we compare the performance of the multi-dimensional diversification approaches under two instantiations: setting the importance of each dimension, $\Pr(dim|Q)$, as proposed in the section entitled Multi-dimensional xQuAD (referred to as Adaptive)

vs. under a uniform distribution assumption (i.e., to 1/3 in this case). We find that the multi-dimensional approaches with the Uniform instantiation does not yield any better performance than their flat versions (except in a few cases). In contrast, the multi-dimensional approaches with the dimensions' importance set using the Adaptive method achieve the best performance and consistently outperform their flat counterparts on all metrics. For most of the metrics, the best performing multi-dimensional diversification method is art_xQuAD, which yields the scores of 0.489, 0.877 and 0.320 for ERR-IA, α-nDCG and P-IA, while its flat counterpart can only achieve 0.477, 0.865 and 0.313, suggesting a relative improvement of 2.6%, 1.4%, and 2.2%, respectively. Note that similar trends are also observed for the DimAware Evaluation, reported in Table 2.4.

Tables 2.5 and 2.6 provide the findings for the approaches based on the supervised R-LTR method (to facilitate comparisons, the results for the art_xQuAD is repeated). Our results reveal that i) our multi-dimensional R-LTR$_{\text{exp}}$ approach (using Adaptive instantiation) with explicit aspects outperforms the baseline R-LTR$_{\text{imp}}$ (which is an implicit diversification method), ii) our implementation of the multi-dimensional R-LTR approach using a two-layer neural network (as in [51]) further improves the performance (since R-LTR$_{\text{expNN}}$ outperforms R-LTR$_{\text{exp}}$), and iii) multi-dimensional R-LTR$_{\text{expNN}}$ yields the best performance only for the ST-Recall metric, while multi-

Table 2.5: Diversification performances of R-LTR using the Flat evaluation. In parentheses, we report the percentage change w.r.t. R-LTR$_{\text{imp}}$. The superscript (*) denotes a statistically significant difference using the Student's paired t-test (at 95% confidence level) w.r.t. R-LTR$_{\text{imp}}$.

		Flat Evaluation				
Div.	Method	ERR-IA	α-nDCG	P-IA	ST-recall	D♯-nDCG
Imp.	R-LTR$_{\text{imp}}$	0.430	0.785	0.282	0.883	0.836
M-Dim Exp.	R-LTR$_{\text{exp}}$	0.435(1.2%)	0.806(2.7%)	0.291(3.2%)	0.933*(5.7%)	0.863*(3.2%)
M-Dim Exp.	R-LTR$_{\text{expNN}}$	**0.461***(7.2%)	**0.849***(8.2%)	**0.305***(8.2%)	**0.958***(8.5%)	**0.897***(7.3%)
M-Dim Exp.	art_xQuAD	0.489	0.877	0.320	0.913	0.903

Table 2.6: Diversification performances of R-LTR using the DimAware evaluation. In parentheses, we report the percentage change w.r.t. R-LTR$_{\text{imp}}$. The superscript (*) denotes a statistically significant difference using the Student's paired t-test (at 95% confidence level) w.r.t. R-LTR$_{\text{imp}}$.

		DimAware Evaluation				
Div.	Method	ERR-IA	α-nDCG	P-IA	ST-recall	D♯-nDCG
Imp.	R-LTR$_{\text{imp}}$	0.450	0.783	0.299	0.891	0.864
M-Dim Exp.	R-LTR$_{\text{exp}}$	0.454(0.9%)	0.798(1.9%)	0.307(2.7%)	0.935*(4.9%)	0.882*(2.1%)
M-Dim Exp.	R-LTR$_{\text{expNN}}$	**0.481***(6.9%)	**0.842***(7.5%)	**0.321***(7.4%)	**0.960***(7.7%)	**0.918***(6.3%)
M-Dim Exp.	art_xQuAD	0.507	0.866	0.337	0.923	0.925

dimensional art_xQuAD performs better for the remaining metrics.

Overall, our experiments confirm a positive answer to our second research question: multi-dimensional approaches with our instantiations are superior to the original diversification algorithms, i.e., the flat versions of xQuAD and PM2, and the baseline R-LTR$_{\text{imp}}$.

To address the last research question of this section, in Table 2.7, we provide the $\alpha-$nDCG results using the automatically extracted aspect sets Auto2015 and Auto2013 for the topic dimension w.r.t. only the Flat evaluation for the sake of brevity.

We find that the trends are generally similar to those observed for the case with the Official aspects, but the actual metric scores are lower (e.g., for multi-dimensional xQuAD, scores are 0.859 and 0.833 for Official and automatic aspects, respectively), as expected. Note that, in contrast to web search result diversification, where automatically extracted aspects usually yield *considerably* lower scores than the official aspects (e.g., [22]), here effectiveness scores are within a closer margin. This might be due to the fact that our automatically extracted aspects always include the main query string too, and the candidate documents in our setup are rather short (as they consist of a title and summary) and hence, less likely to match to noisy terms that may appear due to extraction errors. Furthermore, our application domain is rather

Table 2.7: Diversification performances ($\alpha-$nDCG) of multi-dimensional methods (with the Adaptive Instantiations) using Official vs. automatically extracted aspects (Auto2015 and Auto2013) for the topic dimension.

Div	Method	Official	Auto2015	Auto2013
M-Dim	xQuAD	0.859	0.833	0.830
(Adaptive)	art_xQuAD	0.877	0.840	0.826
	geo_xQuAD	0.860	0.829	0.830

restricted (i.e., in the context of educational search, a given query topic is likely to be associated with a limited set of aspects, which is usually not the case, say, for general web search), and thus, the aspects may be identified accurately from a sufficiently large set of queries. This may also explain why using the Auto2013 aspects, i.e., from a log collected in a different time period, yields only slightly inferior result to using the Auto2015 set, which is from the same log sample as the ground truth.

2.5.2 Click-based Evaluation

In this section, we provide an alternative evaluation based on user clicks. We focus on the second research question of the previous section, i.e., whether multi-dimensional diversification approaches can outperform their flat counterparts in educational search, which lies at the very core of this study.

2.5.2.1 Setup

Query instances and candidate results We use the same query set as before. However, instead of constructing a candidate ranking per query and then diversifying it (as in Section 2.5.1.1), here for each query instance (i.e., an occurrence in the query log), we obtain the result list that has been actually presented to the user, again from the query log, and then diversify the latter, which serves as the candidate ranking in this setup. Note that, for a given query, say, 'triangle', the result lists generated by the underlying retrieval system for different instances are usually quite similar, but there

might be occasional variations due to updates in the document collection and other system-dependent factors. However, the clicked results in each instance may vary widely, as different users may differ in their learning interests for one or more aspects of a given query. The latter type of information, clicks observed for each instance (together with our relevance judgments) are exploited for evaluation in this section: our goal is to re-rank the candidate result list (via diversification) of a given query instance so that the clicked results appear as early as possible in the list (more details are provided later).

Different from the previous section where we had a candidate result set of 25 documents, here we restrict our candidate set to the top-10 documents per query instance, since our evaluation is based on the users' clicks and due to the well-known rank (or position) bias, it is less likely to observe clicks for the documents ranked too low, i.e., after a cutoff value of 10. Overall, for our 40 main queries, we extracted 926 instances together with their top-10 results, which form our dataset for the experiments in this section.

Diversification methods & parameters. In this section, we employ only the flat and multi-dimensional versions of xQuAD since the results from the previous section show them to be representative.

As before, we compute the relevance of the candidate documents to the main query and its aspects in the topic dimension, (i.e., $\Pr(d|q)$ and $\Pr(d|a)$), using BM25. For the (educational) level and (document) type dimensions, we use binary values, as defined in the Multi-dimensional xQuAD section. We estimate the dimension importance probability through Equation (2.5) for the level and type dimensions while we set the importance of topic dimension to 1. For all diversification experiments, we report the results for the best-performing value of the trade-off parameter λ, which is found to be 0.9.

Our evaluation utilizes the Official aspects for the topic dimension. A crucial issue is determining the aspect importance, $\Pr(a|dim, Q)$, for each dimension and its aspects, which was previously assigned a uniform distribution in Section 2.5.1. Since our evaluation in this section is based on user clicks, the diversification algorithm should accurately model the preferences of the user population towards different aspects of

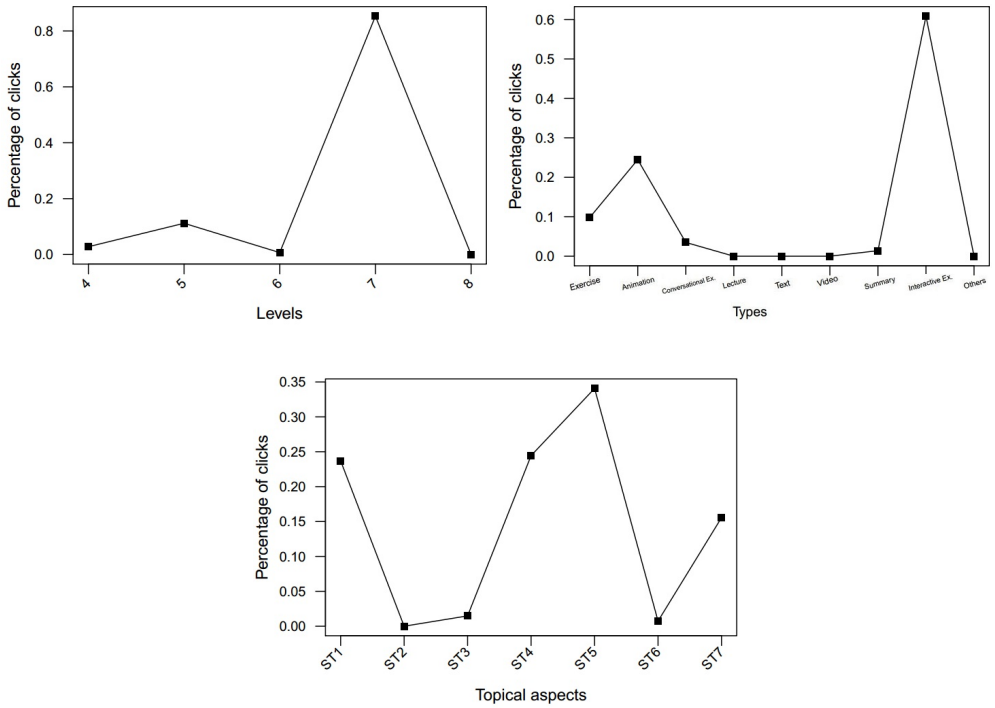


Figure 2.3: Distribution of click counts for the query "light" across each dimension: education level (top-left), type (top-right), topic (bottom). For the latter, the topical aspects shown as ST1 to ST7 on the x-axis correspond to "light and color", "light filter", "white light", "absorption", "refraction", "light year", and "light sources", respectively.

a query, as they may markedly vary a lot. For instance, Figure 2.3 displays the user clicks' distribution over the aspects of each dimension for the query "light". For the education level dimension, the aspect *level 7* is the most popular aspect with a considerably large click-rate i.e., 85.3% of clicks observed over all instances of this query are for the documents covering this aspect. The documents with education levels 4 and 5 are very rarely clicked, while the other levels (6 and 8) are not clicked at all. Similarly, for the topic dimension, documents covering four of the aspects that are identified in the ground truth (namely, *ST1, ST4, ST5, ST7* in Figure 2.3) are clicked often, while the others are neglected.

To illustrate why it is crucial to accurately model the aspect probabilities during diversification, consider the following toy scenario. Assume a query q with three different aspects A, B and C (say, in the topic dimension) and three candidate documents d1, d2 and d3 covering these aspects, respectively. If all aspects are equally likely in the

query log, than top-2 rankings (obtained after diversification) as (d1, d2) or (d2, d3) would be equally good, as each ranking covers two different aspects. However, if we further assume that the users' click rates on the documents covering these aspects A, B, and C are 90%, 5% and 5%, respectively, then it is obvious that a click-based evaluation will favor the ranking (d1, d2) over the ranking (d2, d3), since the majority of its instances, documents covering aspect A will be clicked, yielding a higher evaluation score. This suggests that the top-ranked results should not only cover the diverse aspects, but those diverse aspects that are popular, so that we can improve the click-based metrics.

We learn the aspect probabilities for each query and dimension by splitting our dataset into training and test sets, based on the chronological order. In particular, for each query, we use the first 75% of its instances (in timestamp order) as the training set (adding up to 699 instances) and the rest as the test set (including 227 instances in total). The aspect priors are then obtained from the training set using Equation (2.11):

$$\Pr(a|dim, Q) = \frac{relevant\, clicks\, for\, aspect\, a}{\sum_{a \in (dim,Q)} relevant\, clicks} \tag{2.11}$$

As mentioned before, most users click the top-ranked result(s) regardless of its relevance, a phenomena known as the rank bias. For instance, in our dataset, for about 25% of the instances, only the top-1 or top-2 results are clicked. Naturally, for such rankings, it is almost impossible to improve the click-based metrics via re-ranking (i.e., after the diversification). This is a common issue that arises in the case of conducting a click-based evaluation by re-ranking previously obtained results, usually from a query log. The ideal solution – of conducting an A/B test with the previous and treated rankings – is rarely attainable as the researchers are usually not in control of the underlying retrieval system, which also holds for our case. Hence, following the practice in some previous works [57, 58], we combine the initial ranking with the diversified one (using the well-known Borda Voting method (e.g., see [59]) so that the final ranking is not extremely different from the initial ranking. Furthermore, we always preserve the first document in the initial ranking and apply diversification for the rest of the documents in the list.

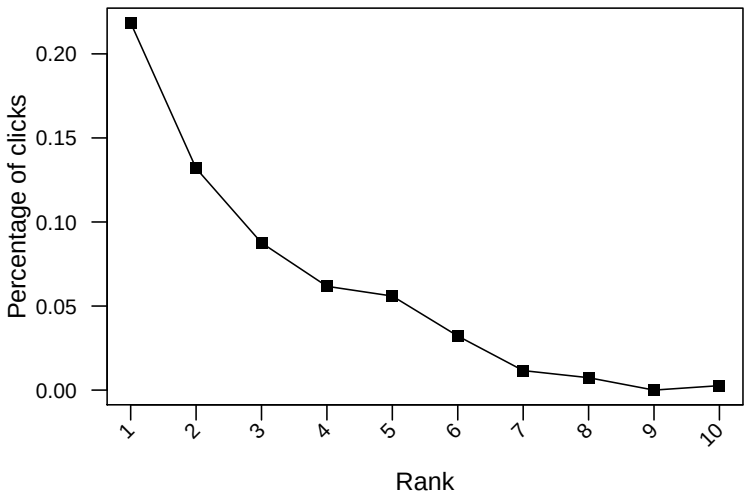


Figure 2.4: Distribution of relevant clicks

In our dataset, clicks distribution can be seen in Figure 2.4. From the Figure 2.4, it can be observed that clicks intensify at top-3 results. We also know that 235 of 926 instances have only one click at top1 or only 2 clicks at top2. In the direction of these information, we therefore verify, as expected, that position bias have a strong effect in our log.

Ground truth & evaluation metrics In this setup, the ground truth is based on the clicked results per query instance, following [58]. Furthermore, we filter them so that only those clicks on the documents that are labeled as relevant in the annotation-based evaluation (cf. Section 2.4.4) are kept in each instance's ground truth[12]. Among the total of 1,895 clicks for all our query instances, only 12% of the clicks are for documents labeled as non-relevant. In other words, 88% of the users' clicks were for documents judged "relevant". For the remaining 12% of clicks, a manual analysis of some randomly chosen results revealed that these clicks are noisy (i.e., the user – most likely to be a young student in this case, as our dataset covers the educational levels in the range of 4 to 8 – may have clicked by mistake or without even checking the snippet) as they seem definitely non-relevant, and hence we discard them.

However, note that not every relevant result may be clicked in every instance; as discussed before, certain users may be interested in certain aspects only, and thus

[12] Note that our annotation study involved the union of all results in all instances of each query, hence, we have a label for every distinct result in each instance.

may skip documents that are labeled as relevant yet covering the other aspects that are not interesting for such users. Therefore, the evaluation framework presented in this section differs from that of Section 2.4.4.

To summarize, for each instance, the ground truth involves those results that are both clicked by the user in this instance's result list and also labeled as relevant by our judges. Based on this ground truth, we compute the traditional relevance metrics as well as the diversification metrics (this is possible, since the ground truth aspects are available for the documents labeled as relevant). We report Precision and nDCG as the traditional metrics, as well their diversification-aware versions, P-IA and $\alpha-$nDCG, at early rank cutoff values of 2 and 5 (Note that we don't report scores @1 as the first result from an initial ranking is always preserved).

2.5.2.2 Results

Table 2.8 presents the diversification performance of the multi-dimensional xQuAD algorithm with different aspect importance weights. The "Uniform" tag in the table denotes that the importances of the aspects under each dimension are assumed to be uniform, whereas the "Priors" tag denotes that aspects' importances of a query across each dimension are learned from the training data and using Equation (2.11). For the multi-dimensional approaches, the dimension importance is set using the Adaptive strategy described in Section 2.3.2.

Table 2.8 reveals that both the flat and multi-dimensional diversification methods (with Priors) outperform the baseline especially for the top-2 results. We also find that the multi-dimensional approach with Uniform aspect priors yield inferior results both to the flat and multi-dimensional approaches (with Priors), and sometimes, even to the non-diversified baseline. This confirms our intuition that the diversification methods in this setup should incorporate realistic aspect priors learned from the user interactions, i.e., clicks. The multi-dimensional approach with the Priors achieves the best results overall, with relative improvements over its flat counterpart reaching up to 2.0% (i.e., 0.350 vs. 0.343) and 1.4% (i.e., 0.442 vs. 0.436) for the diversification and relevance metrics P-IA@2 and P@2, respectively.

Table 2.8: Performances of flat and multi-dimensional xQuAD using the click-based evaluation.

		Relevance				Diversity			
Div	Method	P@2	P@5	nDCG@2	nDCG@5	P-IA@2	P-IA@5	α-nDCG@2	α-nDCG@5
NonDiv		0.411	**0.306**	0.462	0.542	0.345	**0.228**	0.524	0.625
Flat	withPriors	0.436	0.303	0.477	0.544	0.343	0.226	0.532	0.630
M-Dim	Uniform	0.422	0.305	0.461	0.537	0.336	0.224	0.517	0.619
	withPriors	**0.442**	0.305	**0.484**	**0.545**	**0.350**	**0.228**	**0.539**	**0.632**

Table 2.9: Performances of flat and multi-dimensional xQuAD with Priors (macro-averaging over queries).

	Relevance				Diversity			
	P@2	P@5	nDCG@2	nDCG@5	P-IA@2	P-IA@5	α-nDCG@2	α-nDCG@5
NonDiv	0.403	**0.285**	0.449	0.525	0.295	0.210	0.467	0.566
Flat	0.405	0.278	0.452	0.527	0.310	0.213	0.488	0.580
M-Dim	**0.407**	0.282	**0.453**	**0.528**	**0.318**	**0.214**	**0.496**	**0.583**

In our query log, since the number of instances for each query varies (i.e., the minimum and maximum number of instances is 4 and 97, respectively), it is worthwhile to investigate what happens if the diversification scores are first averaged over the instances of each query, and then over the queries (i.e., a macro averaging perspective); so that a query with too many instances does not dominate the overall performance and conclusions drawn.

Table 2.9 presents the diversification performance of the flat and multi-dimensional xQuAD approaches (both with Priors) by macro averaging the scores over queries. The trends are similar to those in Table 2.8, as multi-dimensional xQuAD outperforms both of its competitors, with even larger margins for the diversification met-

rics. In particular, the latter method achieves improvements of 1.6% and 2.6% over its flat counterpart in terms of $\alpha-$nDCG@2 and P-IA@2 scores, respectively. In other words, our gains presented in Table 2.8 still occur when the query frequency effect is eliminated from our evaluation.

Overall, our evaluations based on the user clicks and relevance annotations (as presented in the current and previous sections, respectively) reveal that the proposed multi-dimensional diversification approach yields improvements of up to 2.6% for various relevance and diversification metrics (c.f. Tables 2.3 - 2.9), a finding that indicates the robustness of our approach for the educational search scenario addressed in this study. Given the previous work of [29] that found that a diverse presentation of results yields the highest percentage of users with knowledge gains in comparison to the single- and multi-query searches (measured via various lab-based user studies), we believe that our improvements in diversification performance using the traditional metrics imply a good chance of improving the human learning experience in an educational search context.

2.6 Conclusions

In this chapter, we introduced the multi-dimensional diversification of results in the context of educational search to help the users' learning-oriented search activities. Our proposed enhancement of the xQuAD diversification model (also applied to PM2 and R-LTR) allows the multiple dimensions that are available in this context to be taken into account when ranking documents, such as the type and target educational level of each document. Our extensive experiments upon a newly-created test collection show 2.6% improvement over "flat" diversification approaches and a marked 15.1% improvement over a BM25-based initial ranking obtained within a TREC-style evaluation framework, that is based on relevance annotations, for the ERR-IA metric.

We also employed another evaluation framework based on user clicks. Contrary to the annotation-based evaluation, the click-based setup is sensitive to the users' learning preferences for query aspects, which vary wildly in practice, and hence, the diversification methods use aspect importance priors that are also obtained from the query

logs. In this realistic evaluation framework, multi-dimensional diversification again proves to be useful, for instance, providing good gains of 1.4% and 7.5% for the P@2 metric over the "flat" diversification and non-diversified initial ranking, respectively.

CHAPTER 3

SUPERVISED APPROACHES FOR EXPLICIT SEARCH RESULT DIVERSIFICATION

In this chapter[1], we focus on the explicit diversification methods, which assume that query aspects are known at the diversification time, and leverage supervised learning methods to improve their performance in three different frameworks with different features and goals.

In Section 3.1, we provide a brief introduction and list our major contributions. Next, we review the previous works in the literature. In Section 3.3, we provide preliminaries required in some approaches in areas relevant to our study. Section 3.4 presents our three frameworks for diversification. In Section 3.4.1, we introduce our first framework, LTRDiv, in which we apply typical learning to rank (LTR) algorithms to obtain a ranking where each top-ranked document covers as many aspects as possible. We argue that such rankings optimize various diversification metrics (under certain assumptions), and hence, are likely to achieve diversity in practice. Section 3.4.2 describes our second framework, AspectRanker, where we apply LTR for ranking the aspects of a query with the goal of more accurately setting the aspect importance values for diversification. As features, we exploit several pre- and post retrieval query performance predictors (QPPs) to estimate how well a given aspect is covered among the candidate documents. Lastly, in Section 3.4.3, by the LmDiv framework, we cast the diversification problem into an alternative fusion task, namely, the supervised merging of rankings per query aspect. The experimental setup and results follow in Sections 3.5 and 3.6, respectively. We provide concluding remarks in Section 3.7.

[1] Reprinted in accordance with the copyright conditions of Information Processing and Management, S. Yigit-Sert, I. S. Altingovde, C. Macdonald, I. Ounis, Ö. Ulusoy, Supervised approaches for explicit search result diversification, ©2020, Elsevier. https://doi.org/10.1016/j.ipm.2020.102356.

3.1 Introduction

Diversification is an approach to satisfy the needs of a population of users for ambiguous and/or broad queries, by ensuring that the documents addressing different possible intents of users are surfaced to the top results. Consider the ambiguous query "apple" – the top-ranked documents should cover both possible *aspects* (a.k.a., subtopics or interpretations) of this query, namely, apple as a fruit and the company. For a broad query, say, "machine learning", there is a wide range of aspects, such as the technological aspects (e.g., learning algorithms, code repositories), the social aspects (e.g. ethics, jobs), or the legal aspects (e.g. bias, fairness), which should all be represented – as much as possible – in an unbiased result set. Therefore, given an initial retrieval result for a query, usually called a *candidate set*, diversification methods are applied to generate a final ranking, which lists top-ranked documents that are both relevant and diverse, i.e. covering as many aspects of the query as possible.

Diversification approaches in the literature can be described as implicit or explicit, in how they aim to understand the different possible aspects of a query [6]. The implicit approaches solely exploit the candidate set, i.e., the features of the initially retrieved documents, for diversification. Instead, the explicit approaches assume that the query aspects have been inferred beforehand (say, using topical directories [4] or query reformulations [15]) and aim to prioritize the coverage of these aspects while generating the final ranking. Earlier studies have consistently shown that when the aspects are available, the explicit methods outperform the implicit ones. More recently, explicit diversification methods have also emerged as a promising approach to reduce bias and enhance fairness in various search scenarios (e.g. [13]), where it is reasonable to assume the availability of such query aspects (such as gender, ethnicity and age in a job search scenario [14]). Thus, given the success of explicit diversification methods in typical search scenarios and their usage in new application areas, we argue that exploring ways of further improving their effectiveness is an important and timely research direction.

The key research question tackled in this chapter can be formulated as "How can we exploit supervised learning methods to improve the effectiveness of explicit search result diversification?". To this end, we identify three sub-questions as follows:

- RQ1: How can we employ supervised learning, namely, typical learning to rank (LTR) algorithms, for explicit diversification?
- RQ2: Instead of learning a model for generating a diversified ranking, how can we learn a model to predict the importance of query aspects, to be used in a traditional (unsupervised) diversification method?
- RQ3: How can we cast the diversification problem into a fusion problem, and then adapt supervised learning methods to solve the latter?

To investigate answers to these questions, we build three different frameworks, each of which leverages supervised learning with different features and goals, as follows:

- *LTRDiv:* First, we devise features that capture the aggregated relevance of a candidate document to all query aspects to train a model via typical learning to rank (LTR) algorithms. This framework, LTRDiv, is intended to rank higher the documents relevant to multiple aspects and thus, it aims to maximize the diversity of the ranking (as captured with the intent-aware precision metrics [4], which will be discussed later).
- *AspectRanker:* In our second framework, AspectRanker, we apply supervised learning methods for a sub-task of the explicit diversification process, namely, predicting the importance of the aspects for a user query. While most diversification methods in the literature employ aspect importance as a key component, they usually assume a uniform distribution (i.e., all aspects are equally important) or a popularity-based instantiation obtained from external resources (which may not match the actual representation of aspects in the candidate document set). In our approach, we first re-rank the *candidate documents* for each aspect and employ several pre- and post-retrieval query performance predictors, *QPP*s (e.g., [60]), to estimate how well a given aspect is covered in the candidate set. Then, we train models to rank the aspects for a given query. Finally, we map the aspect rankings to importance values and exploit them in an traditional (unsupervised) diversification method, namely, xQuAD [15].
- *LmDiv:* Inspired by the LambdaMerge (*Lm*) method that aims to combine rankings for different query suggestions [1], we train models that merge rankings

(of candidate documents) for each query aspect. In this case, the trained model (a neural network as in [1]) captures both the relationships of documents and aspects, as well as each aspect's importance, which is again estimated based on QPPs.

3.2 Related Work

In this chapter, we leverage supervised learning to improve the performance of explicit search result diversification, which explicitly models the aspects of a query, and relevance of documents with respect to these aspects is taken into account while diversifying.

The earliest work that devised a technique to exploit known query aspects is IA-Select, proposed by [4], which is similar to its successor xQuAD [15] but it lacked the relevance component in Equation (3.1). The xQuAD method (described in Section 3.3.1) was reported to be the best method across several TREC campaigns, which motivated various optimizations over the original formulation (e.g., [22]). A more recent approach, PM-2, employs a strategy based on the allocation of seats to political parties in elections [42]. Ozdemiray and Altingovde [22] employed the aggregation of rankings (obtained for each query aspect) using unsupervised techniques, such as the well-known CombSum [61]. All of these techniques are unsupervised, as they do not involve any training stage to learn a scoring function for diversification. Instead, the frameworks proposed in our work either directly learn a model (as in LTRDiv and LmDiv) to produce a diversified ranking, or learn a model to predict the aspect importance, which are used in all these prior approaches.

The LambdaMerge method adopted here is introduced by [1] to improve the relevance of query results, and it merges rankings that are obtained over the entire collection for a query and its reformulations. In a follow-up study [62], LambdaMerge is also exploited for the fusion of results obtained over different collections or via different retrieval methods. However, to the best of our knowledge, LambdaMerge has not been applied for merging the re-rankings of the candidate documents for different query aspects, as we propose in this study, for the purpose of diversification. Note

that the LmDiv framework is close to a particular prior work, [22], since both are based on the idea of ranking aggregation; but the latter work employs unsupervised merging methods, whereas the LmDiv framework aims to *learn* a function to merge rankings.

All the aforementioned methods in the literature (namely, IA-Select, xQuAD, PM-2, CombSum based, etc.) require the aspect importance during diversification; however, most of the earlier works assume either a uniform probability distribution [15, 22] or set the aspects' importance using their popularity (say, in a collection [15]). In a recent study, [21] proposed setting the aspects' importance values based on the score of a single QPP. Our AspectRanker framework extends the latter one in several ways: we use several QPPs at the same time as features, and learn a model to produce a ranking of the aspects, which is then mapped to the actual importance values.

Some earlier works showed that the normalization of relevance scores is important for the effectiveness of the unsupervised explicit diversification methods, and proposed alternatives, such as the so-called Virtual [22] and R60 [63] normalization techniques. While we essentially employ the typical sum-based normalization here, our supervised methods may also benefit from incorporating such alternatives which is beyond the scope of this thesis and left as future work.

In the literature, there are various supervised approaches for implicit search result diversification. SVM-DIV [64], which is one of the pioneer works, trains structural support vector machines (SVMs) to learn diverse subsets based on the assumption that if a document has more different words, then it may cover more subtopics. Yet, this work only focuses on diversity and ignores relevance. Liang et al. [65] addressed personalized search result diversification with a supervised approach. They proposed a user-interest topic model to identify the relationships between users and topics, and between the documents and topics. The obtained user-interest features from the topic model are incorporated into a structural SVM as additional constraints so as to be learned to produce both a diverse and personalized ranking list. Wu et al. [66] combined two rankings produced from a traditional LTR model. While the first ranking is the outcome of the model using document-based features (i.e. tf-idf, BM25), and the second ranking uses both document-based features and diversity-biased features that

are extracted over documents retrieved by the former model.

Zhu et al. [17] proposed a Relational Learning to Rank (R-LTR) framework that learns the weights in an equation like MMR that combines relevance score and diversity score between the current document and already selected documents, employing SGD. Xu et al. [67] employed R-LTR as the ranking model and optimized diversity evaluation metrics directly in training. To do so, they optimized an upper bound of the basic loss function using different optimization techniques for bound optimization that yields three learning algorithms, namely PAMM, SGDMM-Log, SGDMM-Exp which use Perceptron [68, 69], stochastic gradient with logistic function, stochastic gradient with exponential function for optimization, respectively.

Xia et al. [70] introduced a model that employs Neural Tensor Network (NTN) to capture the novelty of the document according to already selected documents. NTN composes of a tensor layer, a max pooling layer and a linear layer that compute a novelty score for a document in the candidate set considering selected documents. They attach NTN into ranking models of R-LTR and PAMM.

We are aware of only two works that have attempted to use supervised methods for explicit diversification. In the first one, Zheng et al. [71] proposed a supervised method, L-HSRD, for hierarchical search result diversification. In this method, features are based on the relevance between aspects in each level of hierarchy and the candidate documents, and the model is trained using a sequential selection model (i.e., considering the previous documents in the ranking), as in [17]. Instead, the ranking functions learnt in the LTRDiv and LmDiv frameworks do not apply such a sequential selection process; i.e., they consider each document on its own during training and testing.

In the second work, Jiang et al. [9] proposed a framework deploying a recurrent neural network with an attention mechanism. In particular, while scoring a new document, the attention mechanism emphasizes the aspects that are not covered by the previously selected documents (i.e., following the sequential selection model discussed for L-HSRD). This is again different from the supervised learning employed in our proposed frameworks. In particular, both AspectRanker and LmDiv model the importance of an aspect based on its retrieval quality (over the candidate set) captured via QPPs, while the latter work weighs aspect(s) to “attend" at each iteration based

on the previously selected documents.

3.3 Background and Preliminaries

Our work builds on and/or incorporates various previous approaches, such as explicit diversification (and especially, the xQuAD method), query performance predictors, and supervised algorithms for ranking and merging. In this section, we briefly review the methods that we employ and/or adopt in this work together with the corresponding notations.

3.3.1 Explicit Result Diversification and xQuAD

Consider a query q with a set of known aspects $A_q = \{a_1, \ldots, a_m\}$ and a candidate set D including N documents that is initially retrieved for the main query q. The goal of diversification is to obtain a final ranking R (with $|R| = k$, where k is usually less than N) that is both relevant to the query and diverse (i.e., covering as many and diverse aspects as possible).

One of the most successful explicit diversification approaches is xQuAD [15], which was the top-performer in the Diversity Task of the TREC Web Track between 2009 and 2012 (e.g., see [6]). This is a greedy best-first approach that selects the document $d \in D$ that maximizes Equation (3.1) in each iteration, until k documents are inserted into R.

$$Score(q, d, R) = (1 - \lambda) \Pr(d|q) + \lambda \sum_{a \in A_q} \Big[\Pr(a|q) \Pr(d|a) \prod_{d_j \in R} (1 - \Pr(d_j|a)) \Big], \tag{3.1}$$

In Equation (3.1), $\Pr(d|q)$ and $\Pr(d|a)$ denote the score of a document with respect to the main query, or an aspect, and can be calculated using any effective document ranking approach, such as BM25 [15]. The first summand of Equation (3.1) aims to capture the relevance of a candidate document d to the main query q, while the right hand side represents the diversity, based on the sum of the coverage of each aspect

a ($\in A_q$) by d. In the latter computation, $\Pr(a|q)$ represents the importance of that aspect for the query, and, by default, is uniform across all aspects [15]. Furthermore, xQuAD discounts the score contribution of a document for a particular aspect (i.e., $\Pr(a|q)\Pr(d|a)$) by the probability that the aspect has been already well-covered by the documents selected *earlier* into R, represented by the product term $\prod_{d_j \in R}(1 - \Pr(d_j|a))$. This discounting mechanism aims to prioritize documents with high scores for the aspects that are not yet covered in R, and hence, enhances the novelty of the final ranking. The trade-off parameter λ is used to balance the relevance and diversity in the final ranking R.

In this work, we employ xQuAD as a representative explicit diversification method in our AspectRanker framework (Section 3.4.2) and as a baseline method in our empirical comparative experiments (Section 3.6).

3.3.2 Query Performance Prediction

For a search system, it is important to be able to estimate the quality of the ranking obtained for a query, as it opens the way for several optimizations (such as applying more sophisticated retrieval methods, or suggesting alternative query formulations to the user). Therefore, various query performance predictors (QPPs) have been proposed in the literature (e.g., see [21, 72, 73, 74, 75, 76, 77, 78, 79]). QPPs are broadly categorized as pre-retrieval or post-retrieval with respect to when the estimation can be done, i.e. before or after retrieval. Earlier works reported that both types of predictors are useful and the overall prediction performance may further improve by using different types of QPPs in combination [60].

In this work, we exploit both pre- and post-retrieval QPPs as features to learn a model for predicting the aspect importance values (e.g., $\Pr(a|q)$ in Equation (3.1)) in the AspectRanker and LmDiv frameworks[2]. In the following, we briefly review the QPPs employed to this end. In addition to the notations introduced in the previous section, we follow the notations used in [21], where C and $s_q(d)$ denote the underlying document collection and the relevance score of a document d with respect to query q (i.e.,

[2] QPPs have been exploited as features for other tasks in the context of result diversification, such as classifying the aspect intent [80] and predicting the trade-off parameter λ [81], which are clearly different from the way they are used in our AspectRanker and LmDiv frameworks.

$\Pr(d|q)$ or $\Pr(d|a)$ in Equation (3.1)), respectively; D_q^n represents the top-n ranking of candidate documents D (based on the relevance scores $s_q(d)$) for a query. Note that we employ all the QPPs also for the top-n rankings $D_{a_i}^n$ (where $a_i \in A_q$), i.e., for the ranking of candidate documents w.r.t. each aspect of a query (as will be discussed in Section 3.4).

3.3.2.1 Pre-retrieval Predictors

In this work, we use the following two pre-retrieval predictors:

- **maxSCQ:** This predictor is motivated by the intuition that if the collection contains documents that are similar to the query, then this query is more likely to have a higher performance [77]. The similarity score (SCQ) is computed as:

$$SCQ(q) = \sum_{t \in q}((1 + \ln(freq_{C,t}))\ln(1 + \frac{M}{M_t})) \tag{3.2}$$

  where M is the number of documents in the collection, $freq_{C,t}$ is the frequency of term t in the collection C, and M_t is the number of documents with term t. Zhao et al. where M is the number of documents in the collection, $freq_{C,t}$ is the frequency of term t in the collection C, and M_t is the number of documents with term t. Zhao et al. [77] identified a variant of SCQ called maxSCQ that was the most effective; maxSCQ is the SCQ score for the query term t that maximizes Equation (3.2).

  [77] identified a variant of SCQ called maxSCQ that was the most effective; maxSCQ is the SCQ score for the query term t that maximizes Equation (3.2).

- σ_1**:** Zhao et al. [77] conjectured that as the standard deviation of the query terms' weights increases, the retrieval system would identify relevant documents readily, since documents would discriminate easily. This predictor sums the deviations over the query terms and thus reflects the variability of the query as a whole.

$$\sigma_1(q) = \sum_{t \in q} \sqrt{\frac{1}{M_t} \sum_{d \in C_t} (w_{d,t} - \bar{w}_t)^2} \tag{3.3}$$

$$\bar{w_t} = \frac{\sum_{d \in C_t} w_{d,t}}{|C_t|}, \tag{3.4}$$

where C_t is the set of documents including the query term t while $w_{d,t}$ denotes the weight of query term t in document d (calculated via tf-idf in [77]).

3.3.2.2 Post-retrieval Predictors

In this work, we use various post-retrieval predictors:

- **Weighted Information Gain (WIG):** This QPP estimates retrieval effectiveness by computing the difference between the mean of relevance scores of documents in the ranking D_q^n, and the relevance score of the collection. The intuition behind this approach is that "high quality retrieval should be much more effective than just returning the average document" [78]. The WIG score is calculated as follows:

$$WIG(q) = \frac{1}{n\sqrt{l}}(avg_{d \in D_q^n}(s_q(d)) - s_q(C)), \tag{3.5}$$

where $s_q(d)$ and $s_q(C)$ represent the relevance scores of documents $d \in D_q^n$ and the collection C, respectively. l is the length of the query q, and n is the size of the ranking.

- **Normalized Query Commitment (NQC):** Shtok et al. [73] stated that using the mean of the relevance scores might be misleading, as the ranking may include non-relevant documents. Hence they propose a technique that measures the amount of deviation of the relevance scores of documents in the ranking w.r.t the mean score, and further normalized it w.r.t the collection score.

$$NQC(q) = \frac{\sqrt{\frac{1}{n}\sum_{d \in D_q^n}(s_q(d) - avg_{d \in D_q^n}(s_q(d)))^2}}{s_q(C)} \tag{3.6}$$

- **ScoreAvg:** ScoreAvg[3] has been used as a simpler form of WIG for fusion-based retrieval in [79]. Since they found that normalizing by the query length damages the quality of the prediction when using multiple lists, they employed

sum normalization. That is, the relevance scores, $s_q(d)$, of documents in D_q^n are normalized so that they sum to 1.

- **ScoreDev:** ScoreDev[3] [79], is similar to NQC, but instead of the normalization w.r.t the collection score, it applies sum normalization over the document scores.

Finally, the following three query performance predictors were introduced by [21].

- **ScoreRatio:** This predictor uses the idea that a higher score gap between the first and last documents in a ranking may imply a higher probability of non-relevant documents appearing in this ranking. It is calculated as follows:

$$ScoreRatio(q) = \frac{s_q(d_n)}{s_q(d_1)} \tag{3.7}$$

- **VScoreAvg:** For this QPP, it is assumed that there exists a virtual document that perfectly matches the query, as in [22]. This virtual document d^V would contain only the terms in the query and it would have the average document length in the collection. Then, its score $s_q(d^V)$ is used to normalize the scores $s_q(d)$ (of $d \in D_q^n$) to obtain a predictor based on the mean relevance of the ranked documents, as follows:

$$VScoreAvg(q) = \frac{1}{n} \sum_{d \in D_q^n} s_q^{virtual}(d) \tag{3.8}$$

$$s_q^{virtual}(d) = \frac{s_q(d)}{s_q(d^V)} \tag{3.9}$$

- **VScoreFirst:** The score of the first document in the ranking (after normalization by the score of the aforementioned virtual document) is used as a query performance predictor.

3.3.3 Supervised Learning for Ranking

Modern search engines apply a two stage ranking where first an initial retrieval result is obtained using relatively efficient methods (such as BM25) and then a complex and

[3]The QPP was named this way by [21].

expensive machine learnt ranker is applied over the latter set. To train such specialized models for the ranking task, several algorithms have been introduced in the last two decades (see [82] for an overview).

In a nutshell, the input for a LTR algorithm is an instance vector that is created for each document retrieved for a query, and includes features that capture the query-document matching (e.g., tf-idf, BM25, etc. scores) as well as document-quality features (PageRank, in/out degrees, etc.) and query features (e.g., see [82]). During training, the target label is the graded relevance judgment of a document for the query (usually obtained from the human assessors). The LTR algorithms can consider these labels on their own, in pairs or as a list, giving way to pointwise, pairwise and listwise learning approaches.

In this work, we leverage LTR approaches in two ways: First, in the LTRDiv framework, we cast the diversification problem to a LTR problem with appropriate features and a target feature that is intended to optimize the aspect coverage. Secondly, in the AspectRanker framework, instead of ranking documents, we rank the aspects of a query to infer their importance values. In both cases, we employ two representative LTR algorithms, namely, SVMRank and Random Forests.

3.3.4 Supervised Learning for Result Merging

A well-explored topic in the literature is merging query results that are obtained by using different retrieval methods (e.g. tf-idf, BM25, LTR, etc.) over the same corpora and/or over different collections. In this work, we adopt a particular supervised strategy, LambdaMerge, which has been proposed to merge the rankings that are obtained for a query and its reformulations [1]. Next, we review LambdaMerge and a follow-up variant, and discuss our application of this method to the result diversification problem in Section 3.4.

Assume that a list of top-N results is retrieved for a query and each of its reformulations. The LambdaMerge method consists of two neural networks: the first one, called scoring network, employs features, x_d^r, capturing the relationship between a query reformulation and the document; while the second network, namely a gating

network, uses features, z^r, that represent the quality of each reformulation. The total score of a document is generated by the co-operation of these two networks. The contribution of each reformulation measured by the gating network is multiplied by the score of the document passing through the scoring network, and adding all of them yield the final score of a document. This process is formulated as follows:

$$score_d = \sum_r \psi_r f(x_d^r, \gamma) \tag{3.10}$$

$$\psi_r = \text{softmax}(z^i, \delta) \qquad for\ i = 1, ..., r. \tag{3.11}$$

where r is a query reformulation, ψ represents the quality of the reformulation, which is computed by Equation (3.11), x_d^r is the feature vector of document d over the result list of reformulation r, γ and δ are the weight matrices. f is the key function of the scoring network. While this function can be any differentiable function, Sheldon et al. [1] chose a fully connected two-layer neural network.

Training a neural network that optimizes an evaluation metric (such as Normalized Discounted Cumulative Gain (nDCG) [83]) is a challenge, due to the discontinuity of the metric. Therefore, LambdaMerge has adopted the approach of LambdaRank [84, 85], which overcomes this problem by using a smoothed version of the objective. In particular, the gradients of the objective, namely O, for the parameters of score and gating networks are computed by applying the chain rule as in Equation (3.12):

$$\frac{\partial O}{\partial \gamma_m} = \sum_d \frac{\partial O}{\partial score_d} \frac{\partial score_d}{\partial \gamma_m} \tag{3.12}$$

The most crucial component is the partial derivative of O w.r.t. the score (generated by the output layer of network), $\frac{\partial O}{\partial score_d}$, which can be computed for various information retrieval (IR) evaluation metrics as follows:

$$\frac{\partial O}{\partial score_d} = \sum_i \mid \Delta_{di} \mid (\Upsilon_{d>i} - 1/(1 + e^{score_i - score_d})) \tag{3.13}$$

where i indicates every document in the ranking, and $\mid \Delta_{di} \mid$ is the difference value in the metric when documents i and d are swapped in the ranking; $\Upsilon_{d>i}$ is an indicator that shows which document is more relevant according to the relevance judgements. It is 1 if the document d is more relevant than i, and 0 otherwise. In LambdaMerge, the

authors employ the nDCG metric while computing $\mid \Delta_{di} \mid$ (see [1] for further details). For the other derivatives, i.e., $\frac{\partial score_d}{\partial \gamma_m}$ and $\frac{\partial score_d}{\partial \delta_n}$, the traditional back-propagation scheme is applied, as for the original LambdaRank approach [84, 86, 85] as in Equations 3.14 and 3.15.

$$\frac{\partial score_d}{\partial \gamma_m} = \sum_r \psi_r \cdot \frac{\partial}{\partial \gamma_m} f(x_d^r, \gamma) \tag{3.14}$$

$$\frac{\partial score_d}{\partial \delta_n} = \sum_r \frac{\partial \psi_r}{\partial \delta_n} \cdot f(x_d^r, \gamma) \tag{3.15}$$

$\frac{\partial}{\partial \gamma_m} f(x_d^r, \gamma)$ denotes the original back-propagation in the neural network to update the weights of the scoring network. The back-propagation of the gating network is performed with respect to softmax function to update δ.

Finally note that Lee at al. [62] extended the LambdaMerge architecture by increasing the number of hidden layers in the scoring neural network (as well as injecting additional feature types) to be use in the collection fusion problem which is the task of merging results from isolated and semi-overlapped collections. Following the naming in the latter work, we refer to our diversification framework that is based on the original LambdaMerge as LmDiv-Shallow, while we denote the multi-layer version as LmDiv-Deep.

3.4 Supervised Learning for Explicit Search Result Diversification

In this study, we exploit various supervised learning methods (i.e., LTR algorithms and neural networks) to either generate a diversified final ranking (in the LTRDiv and LmDiv frameworks), or to obtain aspect importance values to be used in combination with a traditional explicit diversification method (in the AspectRanker framework). In the following, we discuss the details for each of these frameworks.

3.4.1 The LTRDiv Framework

In this section, to answer our first research question, RQ1 in Section 3.1, we cast the result diversification problem into a typical ranking problem to exploit the existing LTR algorithms to obtain the final ranking. As discussed in Section 3.3.3, in the traditional setup for LTR, the goal is to maximize the relevance of the ranking for a given query; therefore, during training, the model learns to predict the relevance label of each document for a given query. In this case, the document is represented with the features that capture the query-document matching (e.g., tf-idf, BM25, etc. scores) as well as the document-quality (PageRank, in- and out degree, etc.).

To be able to use a LTR algorithm for diversification, we extend this setup. In particular, we describe both the features and the learning target to capture the relevance of a document not only to the main query, but also to its multiple aspects. To formally define the features and target label, assume a query q (i.e., the main query issued by a user) with a set of known aspects $A_q = \{a_1, \ldots, a_m\}$ and a candidate set D (i.e., the top-N documents retrieved for q). We re-rank the candidate set D for each aspect a_i, using a typical retrieval mechanism (e.g., tf-idf, BM25, etc., as will be discussed later). We denote each such ranking as $D^n_{a_i}$, where $n \leq N$. Intuitively, a document that is ranked at a higher position (i.e., close to the top) in many of these rankings $D^n_{a_i}$ is likely to be relevant to several aspects, and hence, would contribute positively to the diversity of the final ranking. Therefore, we represent a document d's coverage of the aspect set A_q with the features that compute the minimal, maximal and average rank (and score) of d over the rankings $D^n_{a_i}$ for $a_i \in A_q$, shown as follows:

$$f_d = \langle s(d,q), r(d,q), \max_{1\leq i\leq m} s(d,a_i), \frac{1}{m}\sum_{i=1}^{m} s(d,a_i), \min_{1\leq i\leq m} s(d,a_i), \\ \max_{1\leq i\leq m} r(d,a_i), \frac{1}{m}\sum_{i=1}^{m} r(d,a_i), \min_{1\leq i\leq m} r(d,a_i)\rangle \tag{3.16}$$

where $s(d,a_i)$ is the relevance score of document d for aspect a_i (sum-normalized over $D^n_{a_i}$), and $r(d,a_i)$ is the rank of document d in $D^n_{a_i}$. Note that the feature vector f_d also includes $s(d,q)$ and $r(d,q)$, i.e., the relevance score and rank of d for the main query q. In this manner, the feature vector for a document is capable of representing the relevance of a document to both the main query and its aspects. The target label

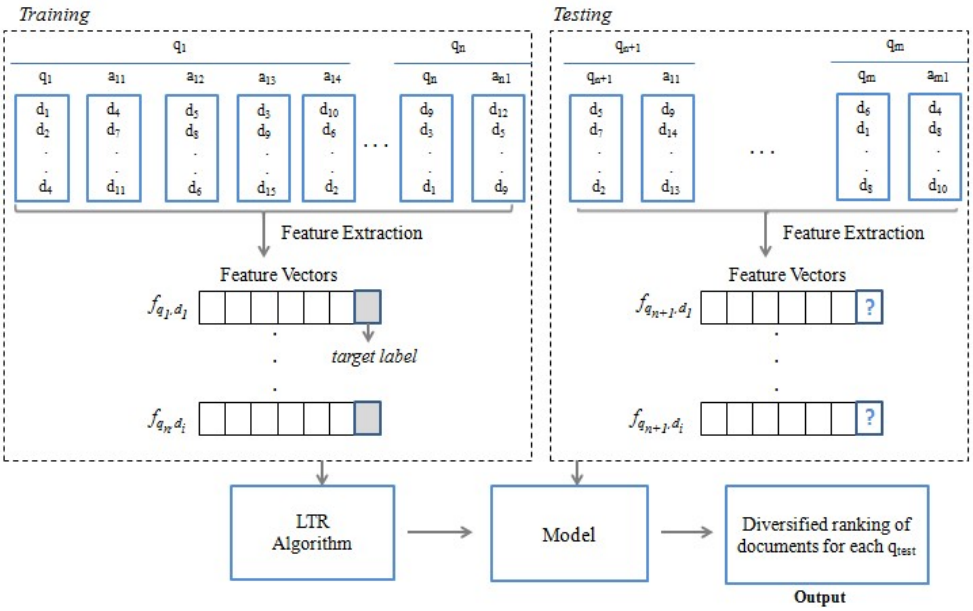


Figure 3.1: LTRDiv framework to obtain a diversified ranking with a typical LTR algorithm.

for a document d is the number of covered aspects, i.e., those a document is stated to be relevant for (with a non-zero grade in the ground truth relevance judgments).

We summarise the LTRDiv framework in Figure 3.1. For the learning component, any LTR algorithm from the literature can be employed, and we discuss the ones employed in this work in Section 3.5. Finally, we justify our choice of the learning target (the number of covered aspects by a document) in this framework by the following observation.

Proposition 1 Assuming all query aspects are equally important and the relevance of each document to each aspect is binary, a ranking of N documents based on the number of covered aspects will yield the optimum scores for the precision oriented intent aware (IA) metrics such as Precision-IA (P-IA) and Discounted Cumulative Gain (DCG)-IA [4], for any cut-off value $k \leq N$.

Example. Let's consider a query q with the candidate set $D_q = \{d_1, d_2, d_3, d_4, d_5\}$. The query has 3 aspects, and the (binary) relevance ($rel_a(d)$) of each document to each aspect is given in Table 3.1. We want to retrieve the top-2 documents for q.

In this case, assuming all aspects are equally important (i.e., $\Pr(a|q) = 1/m$ for a

Table 3.1: Relevance of documents to query aspects for a toy scenario.

	d_1	d_2	d_3	d_4	d_5
a_1	1	1	0	0	1
a_2	0	1	0	0	1
a_3	0	1	0	1	0

query with m aspects), a ranking R of the documents that is based on the covered aspects, namely, $d_2(3), d_5(2), d_1(1), d_4(1), d_3(0)$, maximizes both P-IA and DCG-IA for any rank cut-off value, and hence, the top-2 list includes $\{d_2, d_5\}$. For P-IA, shown in Equation (3.17), this is easy to see, by taking the (constant) multiplication $1/m \times 1/k$ out of the summations and then swapping the order of summations. Then, for each rank position, covering the maximum number of aspects would give the optimal P-IA score, and the ranking $\{d_2, d_5\}$ (covering 3+2 aspects) is optimal. Applying the same transformations for the DCG-IA metric in Equation (3.18), we again see the optimal ranking should provide the highest gains for each possible rank position and hence, for this example, the ranking $\{d_2, d_5\}$ is optimal. Note that, Chapelle et al. [87] also noted that DCG-IA could be optimized by sorting the documents w.r.t. the expected gain, and under the aforementioned assumptions, this means a ranking with respect to the number of covered aspects, as given in Proposition 1. Since the above example conveys the intuition underlying Proposition 1.

$$\text{P-IA@k} = \sum_{a=1}^{m} Pr(a|q) \frac{1}{k} \sum_{j=1}^{k} Rel_a(R_j) \tag{3.17}$$

$$\text{DCG-IA@k} = \sum_{a=1}^{m} Pr(a|q) \sum_{j=1}^{k} \frac{2^{Rel_a(R_j)}}{log(1+j)} \tag{3.18}$$

In light of the above discussion, we argue that training models to predict the number of aspects covered by each candidate document is a meaningful and promising learning target for our LTRDiv framework[4].

[4] We note that the model may underperform if the ground truth for the evaluation includes graded relevance judgments and/or non-uniform aspect importance values; yet in our experiments with graded relevance judgments, LTRDiv was still found to yield a good performance (see Section 3.6).

This also means that LTRDiv is a coverage-based approach and does not take novelty into account, i.e., it neglects the redundancy between the covered aspects. We note that this is a reasonable choice, since an earlier work has shown that solely targeting for the novelty is not effective for diversification [88], while coverage-based methods perform very well comparatively. In our experimental evaluation, we confirm the latter result and show that rankings obtained via LTRDiv yield high diversity scores, not only in terms of the intent-aware metrics, but also w.r.t. those taking novelty into account, such as α-nDCG.

3.4.2 The AspectRanker Framework

In this section, we propose the AspectRanker framework, as an answer to our second research question, RQ2 raised in Section 3.1. In the AspectRanker framework, our goal is to exploit the supervised learning methods, i.e., the LTR algorithms again, to estimate the aspect importance values that are employed in traditional (unsupervised) explicit diversification methods proposed in various earlier works [4, 15, 22, 42]. In Algorithm 1, we specify the overall approach, in three stages, to obtain a diversified ranking. First, we train a model to rank the aspects for a given query. Next, we map each rank to a fixed importance value. Finally, we employ these importance values in a traditional diversification method. In the following, we describe in detail each stage.

For the ranking stage, we need to represent each query aspect with a feature vector and define a target label that would capture the importance of an aspect for a given query. To address this goal, we are inspired by an earlier work [21], which suggests that rather than relying on external resources for inferring aspect importance, one should consider to what extent an aspect is represented in the candidate set. For instance, for a given query “java", if the aspect “java island" is not covered adequately by any of the documents in the candidate set (i.e., all candidates are either relevant to the “programming language" or “coffee" aspects), then assigning uniform importance values to all three aspects (following the common practice in the literature [15]) would not help, but might even mislead the diversification process. To further justify our approach, in Figure 3.2, we provide the percentage of aspects that have a given

Algorithm 1: AspectRanker

Input : Q_{train} : the query set for training

Q_{test} : the query set for test

$\{D_1, .., D_Q\}$: candidate set retrieved for $q_i \in Q_{train} \cup Q_{test}$

$\{A_1, .., A_Q\}$: aspect set for $q_i \in Q_{train} \cup Q_{test}$

Output: R : set of diversified rankings R_i for q_i

for each training query $q \in Q_{train}$ **do**
    for each aspect $a_i \in A_q$ **do**
        Generate the ranking $D^n_{a_i}$ based on $s(d, a_i)$ for $d \in D_q$
        Construct feature vector f_{a_i} = $\langle \text{WIG}(a_i), \text{NQC}(a_i)... \rangle$
        Feed $\langle q, f_{a_i}, target \rangle$ triplets to train a LTR model
    end
end
for each test query $q \in Q_{test}$ **do**
    Obtain the aspect ranking A_q using the LTR model
    for each aspect $a_i \in A_q$ **do**
        Compute the importance of the a_i w.r.t its rank in A_q
    end
    Run xQuAD with computed aspect importance values
end

number of relevant documents in the top-100 candidate set, for the BM25 and TREC runs over 198 queries (described in detail in Section 3.5). Remarkably, a considerable percentage of aspects do not have even a single relevant document retrieved in the candidate set. Similarly, as the plot shows, the number of relevant documents does fluctuate: for a large fraction of aspects there are 1 to 10 relevant documents; but aspects with much larger numbers of relevant documents also exist. Therefore, as in [21], we employ the QPPs to capture the retrieval effectiveness of the top-ranked documents (in the candidate set) for each aspect. However, different from the work of [21], we do not directly employ the estimates of a single QPP as the aspect importance values, but instead learn a model to combine the estimates of several QPPs[5].

[5] Note that the combination of QPPs has been explored independently in earlier works [60], but we are not aware of any application in the context of result diversification for aspect ranking.

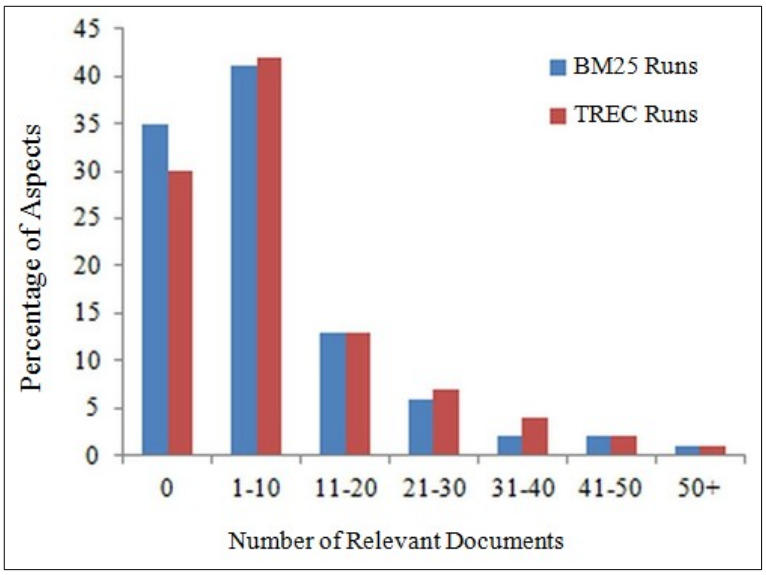


Figure 3.2: Percentage of aspects (y-axis) with a given number of relevant documents (x-axis) in the candidate document sets (BM25 and TREC runs) for 198 queries.

Formally, following the notations in the previous section, we again obtain the re-rankings $D^n_{a_i}$ of the candidate set D for each aspect a_i of a query q. For each such ranking, we compute the QPPs described in Section 3.3.2, namely, WIG, NQC, ScoreAvg, ScoreDev, ScoreRatio, VScoreAvg, VScoreFirst, maxSCQ, and σ_1. As the target label for each aspect, we calculate the well-known Precision metric over $D^n_{a_i}$, as follows:

$$Precision_{a_i,D} = \frac{|D^n_{a_i} \cap Rel_{a_i}|}{n} \tag{3.19}$$

where n ($\leq N$) is the size of the ranking for a_i and Rel_{a_i} is the set of documents judged relevant for the aspect a_i of the given query. Note that such a target label is an approximation, i.e., it is not guaranteed that using the target value as the aspect importance would maximize the diversification performance. However, the other alternative, trying all importance values (within a given range) for all aspects is prohibitively expensive, especially for queries with more than a few aspects. Hence, we opt to employ Equation (3.19) as a proxy for the aspect importance to be predicted. These training instances are then fed to a LTR algorithm to learn a ranker for aspects.

During testing, once we obtain a ranking of aspects for a given query, we apply a simple procedure to map the ranks to actual importance values. We assign an aspect

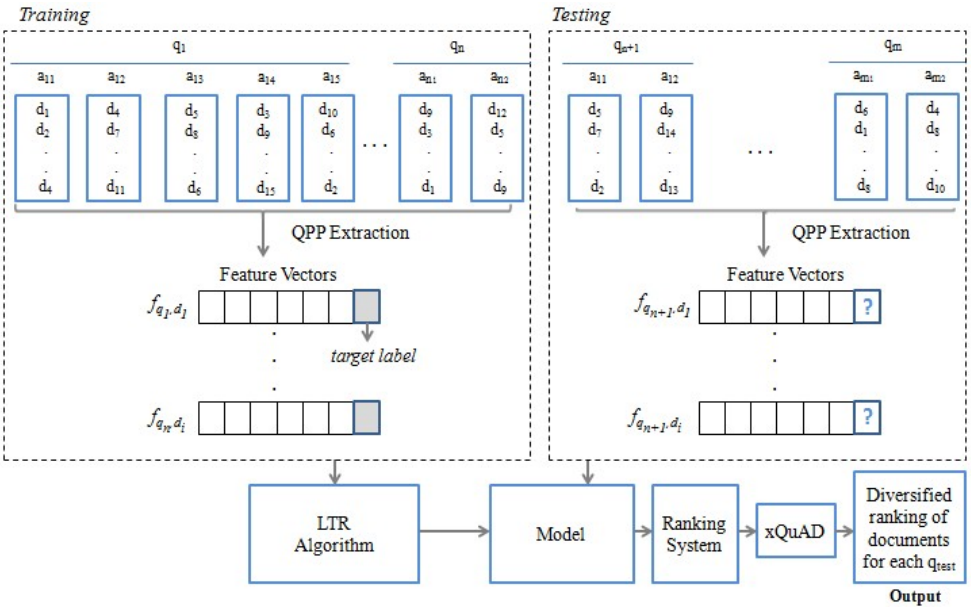


Figure 3.3: AspectRanker framework to obtain a diversified ranking with a typical LTR algorithm.

the importance value that is inversely proportional to its rank, i.e., for m aspects, the top-ranked one has the importance value m and the last one has 1. Note that one could also train a model, say using regression, to predict the $Precision_{a_i,D}$ value directly, but as discussed above, the target value in the model is a proxy for the actual aspect importance and hence, ranking aspects may produce more generalizable results than trying to predict an exact importance value. In our reported evaluation, we show that such an approximation yields indeed a very good performance. Finally, we use the estimated aspect importance values (after sum normalization) in an explicit diversification method to obtain a diversified ranking for a test query. The overall AspectRanker framework is presented in Figure 3.3.

3.4.3 The LmDiv Framework

Our third research question, RQ3 raised in Section 3.1, is addressed in this section. We tailor the LambdaMerge approach [1] to search result diversification casting the diversification problem as a fusion task, namely, the supervised merging of rankings per *query aspect*. In one sense, this approach brings together the previous two frameworks, as the model is based on the features that capture document-aspect relevance

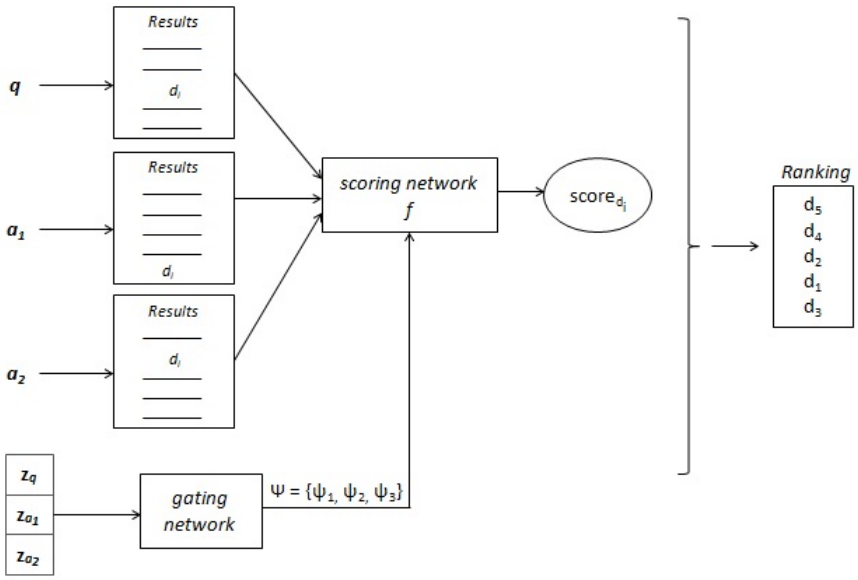


Figure 3.4: The architecture of LmDiv (based on [1]).

as well as those that capture the importance of each aspect. Similar to LambdaMerge (reviewed in Section 3.3.4), our LmDiv framework simultaneously trains two neural networks, a scoring network that creates a score for each document-aspect pair, and a gating network that generates an importance value for each aspect (Figure 3.4). The final score of a document is the weighted sum of the document-aspect scores with the corresponding aspect importance values, computed as follows:

$$score(d) = \sum_{a_i \in A} \psi_{a_i} h(f_{d,a_i}, \gamma) \tag{3.20}$$

$$\psi_{a_i} = \text{softmax}(z_{a_i}, \delta) \qquad for\ a_i \in A. \tag{3.21}$$

where a_i is a query aspect, ψ is the aspect importance, which is computed by Equation (3.21), γ and δ are the weight matrices and $h()$ is the scoring neural network. In the following, we describe f_{d,a_i}, the feature vector of document d for aspect a_i, and z_{a_i}, the feature vector of aspect a_i, in detail.

Different from the LTRDiv framework, in this case, the feature vectors (f_{d,a_i}) are created for each $\langle d, a_i \rangle$ pair where $d \in D$, and $a_i \in A$ for a given query. Note that, in this case, we treat the original query q as an aspect and create all these features for q, as well. As before, $D^n_{a_i}$ denotes a ranking of candidate documents w.r.t. a_i. Then, following the practice in [1], we compute the following features for $\langle d, a_i \rangle$:

- $s(d, a_i)$: The raw relevance score of the document d for a_i
- $r(d, a_i)$: The rank of the document d in $D^n_{a_i}$
- SumNormScore: The relevance score $s(d, a_i)$ after applying sum normalization over all $d \in D^n_{a_i}$
- VirtualNormScore: The relevance score after applying virtual normalization (see Section 3.3.2 for the description of VScoreAvg [21]).
- StandardScore: The relevance score after applying z-score normalization.
- IsInTop: A binary indicator denoting whether the document is in $D^n_{a_i}$ (i.e., when $n < N$, some candidate documents would not appear among the top-ranked documents for a_i).

For the gating network, we again represent each aspect a_i using a feature vector, z_{a_i}, based on QPP scores over $D^n_{a_i}$, namely, WIG, NQC, ScoreAvg, ScoreDev, ScoreRatio, VScoreAvg, VScoreFirst, maxSCQ, and σ_1, as in the previous section. Hence, the network would learn the aspect importance values, which reflect the quality of each aspect's representation in the candidate result set.

During training, for each candidate document and aspect pair (d, a_i), the feature vectors f_{d,a_i} and z_{a_i}, are obtained and fed *simultaneously* to the scoring and gating networks, respectively, for each aspect a_i of the query; and their results are combined to obtain the final document score using Equation (3.20). Once all documents for a query are scored, the LambdaRank gradients of the objective function is computed for the back-propagation.

The original LambdaMerge method aims to optimize the nDCG metric, as shown in Equation (3.13), where the ground truth is simply based on the query-document relevance judgments. In this work, we define a different representation of the ground truth that is more appropriate for the diversification task. As in Section 3.4.1, for each document, we use the number of covered aspects as its target label. In this case, the $\Upsilon_{d>i}$ parameter in Equation (3.13) returns 1 when a document d is relevant to a *larger* number of aspects than document i. Note that, such a formulation would learn a model to generate rankings in descending order of the number of covered aspects

per document, as in the case of the LTRDiv framework, albeit using different features for the documents and aspects, and a neural network model optimizing a list-wise metric, i.e., nDCG. In this sense, LMDiv is also a coverage-based approach w.r.t. the categorization in [80].

As a further extension, we modify Equation (3.13) to allow direct optimization of a metric specifically proposed for evaluating diversity, namely, nDCG-IA [4]. In our adaptation, we compute the impact of swapping two documents for each aspect, separately, and then average over the aspects, as follows:

$$\frac{\partial O}{\partial score_d} = \sum_a \Pr(a|q) \sum_i \mid \Delta_{di} \mid (\Upsilon_{d>i} - 1/(1 + e^{score_i - score_d})) \tag{3.22}$$

In Equation (3.22), $\Pr(a|q)$ denotes the aspect importance in the *ground truth* (if available), i indicates every document in the ranking, $\mid \Delta_{di} \mid$ is the difference value in the nDCG metric when documents i and d are swapped in the ranking, and $\Upsilon_{d>i}$ is an indicator that shows which document is more relevant (to an aspect a) according to the relevance judgements (which may be binary or graded). It is 1 if the document d is more relevant than i, and 0 otherwise. Finally note that, any target metric used in this context must be *consistent* [89], i.e. an improving swap (where a document with a higher label moves above one with a lesser label) must result in a positive or 0 change in the metric. nDCG-IA is also usable as the training objective for LmDiv, as it reduces to nDCG per aspect (captured in the inner summation of Equation (3.22)) and nDCG is known to be consistent (e.g., [89]).

3.5 Experimental Setup

Dataset and runs. We employ query (topic) sets that have been developed for the the Diversity Task of the Text REtrieval Conference (TREC) Web Track between 2009 and 2012. Each set includes 50 queries (except for 2010, which has 48) along with their aspects and relevance judgments at the query and aspect levels. The candidate sets (i.e., the initial retrieval results per query, or shortly, *run*) are obtained over the ClueWeb09 Category B dataset, which consists of about 50 million English

web pages [18]. For all diversification methods (those proposed and the baselines), we employ the official query aspects to isolate the evaluation of the diversification method from that of the aspect inference stage (as in several previous works, such as [15, 42, 22]).

We present the performance of the proposed frameworks on two types of runs. `BM25 runs` are obtained by processing each query over the collection using our own retrieval system implementing the traditional BM25 model. We opt for BM25 because it is still the most widely used IR model in practical settings, (e.g., as a ranking feature in Yahoo [90]), and it does not require additional features (as in the LTR algorithms), enabling others to replicate our experiments. The parameters of BM25, namely k_1 and b, are set experimentally to 1.2 and 0.5, respectively. The documents with a spam percentile-score of 60 or lower according to the Waterloo Spam Rankings [6] are eliminated from the results, following [91]. We retrieve the top-100 documents for each query as the candidate set (i.e., N=100). Whenever needed, we apply sum normalization over the BM25 scores.

To be able to evaluate the diversification performance when the initial retrieval stage employs more sophisticated approaches beyond BM25, we also select the best runs submitted to previous TREC campaigns, which we refer to as `TREC runs`. While doing so, we only consider runs submitted to the ad hoc track (i.e., without any diversification method applied) over the ClueWeb09 Category B collection (following the practice in [92, 93]). The best performing run is the one that yielded the highest α-nDCG@20 score for a given year. The ids of the selected runs for each year are as follows: Ucdsiftinter (2009), uogTrB67 (2010), Srchvrs11b (2011) and Qutparabline (2012). As in the previous case, we focus on the top-100 results from each run for diversification.

Note that, for the TREC runs, we do not have access to the retrieval methods employed to generate each of these runs. Therefore, for $s(d, q)$, i.e., the relevance score of a candidate document d for the main query q, we employ the score provided in the corresponding run (after normalization). To compute $s(d, a_i)$, the relevance of a document to an aspect a_i, we use BM25, following the practice in [92, 93].

[6] `http://plg.uwaterloo.ca/gvcormac/clueweb09spam/`

LTR algorithms. For the LTRDiv and AspectRanker frameworks, we experiment with two LTR algoritms, namely, SVMRank[7] and Random Forests (RF). The former is a well-known pairwise LTR algorithm that optimizes the pairwise-loss over the training instances. We choose the second algorithm, RF, as a representative for the RankLib software package[8], as it has been shown to be the best LTR method among various competitors in a recent study [94]. Our preliminary experiments also revealed that it is the best performing RankLib method in our setting.

Setup for the supervised learning frameworks. For all supervised methods, we apply 5-fold cross-validation to evaluate the performance. For LTRDiv, we train the LTR algorithms using the top-100 candidate documents for the training queries. In AspectRanker, we calculate the aspect features (i.e., QPPs) using the top-20 documents of the re-rankings, i.e., set $n = 20$ for D_q^n (since [21] suggested that considering only the top-ranked documents is adequate to determine the aspect representation quality in the candidate set). The diversification stage in AspectRanker employs xQuAD [15], which is applied again over a candidate set of 100 documents. Finally, the LmDiv framework is trained with the re-rankings of the top-25 candidate documents per query aspect.

For the *shallow* version of the LmDiv framework, we train a fully connected two-layer neural network with four neurons in the hidden layer and a linear combination function as the output, as in [1]. The multi-layer version (referred to as *deep* following [62]) employs four hidden layers, each with fifteen neurons. Both neural networks are trained by stochastic gradient descent. In our preliminary experiments, the number of epochs and learning rate are determined as 25 and $5{\cdot}10^{-3}$, respectively, over the BM25 run for the TREC 2010 topic set, and the same values are employed for all the other runs and topic sets. We supply the training queries in a random order for each epoch. We batch the parameter updates by query for faster training as in [84]. The neural networks optimize the nDCG and nDCG-IA metrics, as discussed in Section 3.4.3.

Baselines. In addition to reporting performance for the non-diversified (NonDiv) ranking, we apply two strong baselines. First, we use xQuAD with uniform aspect

[7] http://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html
[8] https://sourceforge.net/p/lemur/wiki/RankLib/

importance values[9] (as commonly employed in the literature [15, 42, 22]. Secondly, we employ an xQuAD variant where the aspect importance values are based on the estimations of a single QPP. In particular, we use the variant that employs the ScoreRatio predictor (see Section 3.3.2), as it is found to be the best-performing one in [21]. We refer to the latter baseline as xQuAD_{SR}. For both of these baselines, the probabilities $\Pr(d|q)$ and $\Pr(d|a)$ are based on the respective sum-normalized relevance scores $s(d, q)$ and $s(d, a)$, of which computations are specified where the BM25 and TREC runs are described; and the trade-off parameter λ is determined using a 5-fold cross-validation.

Evaluation. We report results in terms of the widely used diversification metrics (see [6] for an overview), namely, ERR-IA [19], α-nDCG [20], Precision-IA [4], ST-recall [36], and MAP-IA [4], at the cut-off value of 20. Furthermore, we provide a more detailed picture for the α-nDCG metric at different cut-off values (2, 10 and 20), since this metric is capable of assessing both diversity and novelty in the results. Note that while certain methods try to learn and exploit aspect importance during the diversification stage, the evaluation stage assumes that all aspects are equally important, which is the common practice in the literature. All metrics are computed using the ndeval software[10]. We use the Student's two-tailed paired t-test (at 95% confidence level) for analyzing statistical significance.

3.6 Experimental Results

In this section, we first provide the evaluation results for all three frameworks using the BM25 runs. Next, for the best performing framework, LmDiv, we provide further results using the TREC runs.

[9] Note that, in our preliminary experiments, we also employed an alternative, popularity-based importance values (computed using the estimated number of results for each aspect from a major search engine) as in [15], and verified that uniform values yield superior results.

[10] `http://trec.nist.gov/data/web10.html`

Table 3.2: Diversification performance of the LTRDiv framework for the BM25 runs (over TREC 2009-2012 topic sets). The superscripts with (†) and (∗) denote a statistically significant difference from NonDiv and xQuAD at 0.05 level, respectively. For LTRDiv variants, % gains w.r.t. $xQuAD_{SR}$ are shown in parentheses.

	BM25 Runs				
Method	ERR-IA@20	α-nDCG@20	P-IA@20	ST-Recall@20	MAP-IA@20
NonDiv	0.2626	0.3567	0.1558	0.5705	0.0351
xQuAD	0.3061	0.4089	0.1751	0.6199	0.0447
$xQuAD_{SR}$	**0.3309**	**0.4294**	0.1813	**0.6211**	0.0462
$LTRDiv_{SVM}$	$0.3188^{\dagger}$(-3.7%)	$0.4183^{\dagger}$(-2.6%)	$\mathbf{0.1908}^{\dagger,*}$(5.2%)	0.6146(-1.0%)	$\mathbf{0.0491}^{\dagger,*}$(6.3%)
$LTRDiv_{RF}$	$0.2984^{\dagger}$(-9.8%)	$0.4020^{\dagger}$(-6.4%)	$0.1812^{\dagger}$(-0.1%)	0.6168(-0.7%)	$0.0430^{\dagger}$(-6.9%)

3.6.1 Diversification Performance of Supervised Learning for the BM25 Runs

LTRDiv Framework. We begin with presenting the performance of our first framework, LTRDiv, presented in Section 3.4.1. Table 3.2 reports the performance of LTRDiv with two typical LTR algorithms, i.e., SVMRank and RF, in terms of diversity-aware metrics. As a first observation, we see that LTRDiv provides a notable improvement over the non-diversified BM25 baseline for all metrics. For instance, while NonDiv yields an α-nDCG score of 0.3567, LTRDiv with SVMRank yields 0.4183 and with RF it yields 0.4020.

For the majority of metrics, we observe that using SVMRank in LTRDiv is better than using RF. LTRDiv with SVMRank also outperforms the xQuAD baseline for all metrics except ST-recall, and performs better than the most-effective baseline, $xQuAD_{SR}$, for the P-IA and MAP-IA metrics. In the latter case, the improvements w.r.t. P-IA and MAP-IA metrics reach up to 5.2% and 6.3%, respectively, over $xQuAD_{SR}$. In short, we conclude that LTRDiv is better than NonDiv and a strong diversification baseline, xQuAD; but it can beat the strongest baseline, $xQuAD_{SR}$, for only two of the evaluation metrics. These results indicate that taking the aspect importance values into account is important for diversification (as the most effective approach in Table 3.2, $xQuAD_{SR}$, employs the ScoreRatio predictor for this purpose), and justify our mo-

tivation for the AspectRanker and LmDiv frameworks, both of which aim to predict such importance values via supervised learning.

AspectRanker Framework. Table 3.3 presents the performance of our second framework, AspectRanker, where our goal is to predict the aspect importance values to be used in a traditional (unsupervised) diversification algorithm, namely, xQuAD. Again, we employ the SMVRank and RF algorithms with AspectRanker. Our findings show that AspectRanker, with any of these LTR algorithms, outperforms (significantly) both the NonDiv and xQuAD baselines for all metrics (yielding relative gains of up to 26% (0.2626 $\rightarrow$ 0.3308) and 8% (0.3061 $\rightarrow$ 0.3308), over NonDiv and xQuAD, respectively, for ERR-IA). We also observe that employing SVMRank in AspectRanker yields a better performance than employing RF, for most of the metrics. More crucially, AspectRanker (with SVMRank or RF) can also beat the strongest baseline, xQuAD_{SR}. Specifically, AspectRanker (with SVMRank) yields relative improvements of 0.8%, (0.4294 $\rightarrow$ 0.4328), 2%, (0.1813 $\rightarrow$ 0.1849), and 1.7%, (0.6211 $\rightarrow$ 0.6314) and 4.3% (0.0462 $\rightarrow$ 0.0482) for α-nDCG, P-IA, ST-Recall and MAP-IA metrics, respectively, in comparison to xQuAD_{SR}. These findings indicate that the supervised learning of aspect importance values provides higher performance improvements to the xQuAD method, in comparison to estimating these importance values using a single estimator, as in xQuAD_{SR}.

LmDiv Framework. In our last framework, LmDiv, we evaluate the impact of exploiting supervised learning for both determining the aspect importance values and generating the final document scores for ranking, simultaneously. In Table 3.4, we report the diversification performance of deep and shallow neural network architectures (described in the experimental setup), referred to as LmDiv-Shallow and LmDiv-Deep, respectively. Note that we experimented with optimizing both of the nDCG & nDCG-IA metrics (as discussed in Section 3.4.3) for both cases; and found out that the results with the former metric are slightly better. As a consequence, we only report the latter results for the sake of brevity.

Table 3.4 shows that LmDiv-Shallow outperforms the baselines for all metrics but MAP-IA (for which the same score is obtained as xQuAD_{SR}), albeit with relative improvements less than 2%. LmDiv-Deep is not as effective as the diversification base-

Table 3.3: Diversification performance of the AspectRanker framework for the BM25 runs (over TREC 2009-2012 topic sets). The superscripts with (†) and (∗) denote a statistically significant difference from NonDiv and xQuAD at 0.05 level, respectively. For the AspectRanker variants, % gains w.r.t. xQuAD_{SR} are shown in parentheses.

	BM25 Runs				
Method	ERR-IA@20	α-nDCG@20	P-IA@20	ST-Recall@20	MAP-IA@20
NonDiv	0.2626	0.3567	0.1558	0.5705	0.0351
xQuAD	0.3061	0.4089	0.1751	0.6199	0.0447
xQuAD_{SR}	**0.3309**	0.4294	0.1813	0.6211	0.0462
$\text{AspectRanker}_{\text{SVM}}$	$0.3308^{\dagger,*}$(0.0%)	$\mathbf{0.4328}^{\dagger,*}$(0.8%)	$0.1849^{\dagger,*}$(2.0%)	$\mathbf{0.6314}^{\dagger}$(1.7%)	$\mathbf{0.0482}^{\dagger,*}$(4.3%)
$\text{AspectRanker}_{\text{RF}}$	$0.3297^{\dagger,*}$(-0.4%)	$0.4320^{\dagger,*}$(0.6%)	$\mathbf{0.1852}^{\dagger,*}$(2.2%)	$0.6290^{\dagger}$(1.3%)	$0.0470^{\dagger}$(1.7)%

Table 3.4: Diversification performance of the LmDiv framework for the BM25 runs (over TREC 2009-2012 topic sets). The superscripts with (†) and (∗) denote a statistically significant difference from NonDiv and xQuAD at 0.05 level, respectively. For the LmDiv-S (Shallow) and LmDiv-D (Deep) variants, % gains w.r.t. xQuAD_{SR} are shown in parentheses.

	BM25 Runs				
Method	ERR-IA@20	α-nDCG@20	P-IA@20	ST-Recall@20	MAP-IA@20
NonDiv	0.2626	0.3567	0.1558	0.5705	0.0351
xQuAD	0.3061	0.4089	0.1751	0.6199	0.0447
xQuAD_{SR}	0.3309	0.4294	0.1813	0.6211	**0.0462**
LmDiv-S	$0.3344^{\dagger,*}$(1.1%)	$0.4358^{\dagger,*}$(1.5%)	$\mathbf{0.1845}^{\dagger,*}$(1.8%)	$0.6226^{\dagger}$(0.2%)	$\mathbf{0.0462}^{\dagger}$(0.0%)
LmDiv-D	$\mathbf{0.3454}^{\dagger,*}$(4.4%)	$\mathbf{0.4410}^{\dagger,*}$(2.7%)	$0.1748^{\dagger}$(-3.6%)	$\mathbf{0.6250}^{\dagger}$(0.6%)	$0.0442^{\dagger}$(-4.3%)

lines for P-IA and MAP-IA, but yields the highest scores for the ERR-IA, α-nDCG and ST-Recall metrics; providing relative improvements of 12.8%, 7.8%, 0.8% over xQuAD, and 4.4%, (0.3309 $\rightarrow$ 0.3454), 2.7%, (0.4294 $\rightarrow$ 0.4410), 0.6%, (0.6211 $\rightarrow$ 0.6250) over xQuAD_{SR}, respectively. Note that the gains of the LmDiv-Deep method over xQuAD (and also over NonDiv) are statistically significant. In comparison to

xQuAD_{SR}, the gains are not identified as significant using a paired t-test, yet they are still numerically impressive, i.e., up to 4.4%.

We also note that the LmDiv-Deep approach is the overall winner for the ERR-IA and α-nDCG metrics (cf. Tables 3.2, 3.3 and 3.4). These two metrics address both the diversity and novelty of a ranking (as they have the diminishing return property, i.e., submodularity) and hence, they are seen as a better fit for assessing diversification effectiveness [87]. Therefore, in what follows, we choose to provide results for α-nDCG, as a representative for the family of submodular metrics – which are used extensively in earlier works as well as in the Diversity Task of the TREC campaigns – at additional cut-off values for further insights.

Table 3.5 demonstrates the diversification performance of the LmDiv-Shallow and LmDiv-Deep methods for the α-nDCG evaluation metric at cut-off values of 2, 10 and 20 (the last column is repeated from Table 3.4 to facilitate comparisons). We see that both versions of LmDiv (i.e. Shallow or Deep) in Table 3.5 outperform xQuAD and xQuAD_{SR} at all rank cut-off values. In particular, LmDiv-Deep achieves the best performance and provides a relative improvement of 7.2%, (0.3444 $\rightarrow$ 0.3693), 3.2%, (0.3997 $\rightarrow$ 0.4125) and 2.7% (0.4294 $\rightarrow$ 0.4410) for α-nDCG@2, 10, and 20, respectively, over the strongest baseline, xQuAD_{SR}.

Next, we provide a gain/loss analysis to identify under what circumstances the LmDiv approach is more useful, i.e., provides gains over the baselines xQuAD and xQuAD_{SR}. To this end, since LmDiv aims to learn aspect importance values based on the evidence in the candidate document set D, we partition the query set according to the coverage of aspects there, as in [42]. Specifically, we compute the ST-Recall scores over the BM25 runs (for our 198 queries), and split them into two groups, i.e., the queries for which the candidate set covered more than 50% of their aspects, and those covering less than 50%.

In the top two rows of Table 3.6, for each of these ST-Recall ranges, we present the percentage of queries that LmDiv (Shallow or Deep version) helps and hurts (shown as "Impr. Q."and "Hurt Q.", respectively), in terms of the α-nDCG scores, with respect to our first baseline, xQuAD. For each group of queries, we also present the overall improvement of the α-nDCG score, again over xQuAD (i.e., to show the total

Table 3.5: Diversification performance (at different rank cut-off values) of the LmDiv framework for the BM25 runs (over TREC 2009-2012 topic sets). The superscripts with (†) and (∗) denote a statistically significant difference from NonDiv and xQuAD at 0.05 level, respectively. For the LmDiv variants, % gains w.r.t. xQuAD_{SR} are shown in parentheses.

	BM25 Runs		
Method	α-nDCG@2	α-nDCG@10	α-nDCG@20
NonDiv	0.2563	0.3214	0.3566
xQuAD	0.3117	0.3776	0.4089
xQuAD_{SR}	0.3444	0.3997	0.4294
LmDiv-S	$0.3484^{\dagger,*}(1.2\%)$	$0.4067^{\dagger,*}(1.7\%)$	$0.4358^{\dagger,*}(1.5\%)$
LmDiv-D	$\mathbf{0.3693}^{\dagger,*}(7.2\%)$	$\mathbf{0.4125}^{\dagger,*}(3.2\%)$	$\mathbf{0.4410}^{\dagger,*}(2.7\%)$

Table 3.6: The percentage of queries improved and hurt (in terms of α-nDCG) by the LmDiv variants over the baselines, xQuAD and xQuAD_{SR}, when queries are grouped by ST-Recall of the initially retrieved documents (BM25 runs over TREC 2009-2012 topics). The Score Impr. column presents the relative α-nDCG score improvement w.r.t. the corresponding baseline.

ST-Recall ranges	[0-0.5)			[0.5, 1]		
No of queries	65			133		
	Impr. Q	Hurt Q	Score Impr.	Impr. Q	Hurt Q	Score Impr.
LmDiv-S vs. xQuAD	32.31%	27.69%	5.14%	62.41%	34.59%	6.81%
LmDiv-D vs. xQuAD	30.77%	30.77%	2.85%	63.16%	34.59%	8.64%
LmDiv-S vs. xQuAD_{SR}	32.31%	27.69%	2.85%	52.63%	45.86%	1.27%
LmDiv-D vs. xQuAD_{SR}	26.15%	33.85%	0.61%	54.89%	42.86%	3.01%

effect of the improved and hurt queries on the performance).

Our results show that, generally, a larger percentage of queries are improved than being hurt by LmDiv. We further observe that the percentages of both improved and hurt

queries increase as the ST-Recall goes up. However, the increase for the improved queries is much higher, i.e., using our best performing LmDiv-Deep approach, the percentage of improved queries against xQuAD goes from 30.77% to 63.16% (more than doubled), while the percentage of hurt queries shows a small increase (from 30.77% to 34.59%). We also observe that for the queries with higher ST-recall, the score gains are higher (e.g., again for LmDiv-Deep, the relative score improvements against xQuAD is 2.85% for low-recall queries and 8.64% for high-recall queries). These findings indicate that the LmDiv approach is more useful when the candidate result set covers a reasonable number of query aspects.

In Table 3.6, the bottom two rows present a similar analysis against xQuAD$_{SR}$. Since the latter is a stronger baseline, the percentage of improved queries is lower for both query groups (in comparison to the xQuAD case), but the trend is similar, i.e., for LmDiv-Deep, the percentage of improved queries is again more than doubled (from 26.15% to 54.89%) comparing the low- and high-recall ranges.

Another question we seek to answer in this section is the impact of QPPs in the trained models, since several QPPs are employed as features to represent the aspects for the gating network component of the LmDiv framework (Figure 3.4). In Table 3.7, we present the QPP features' weights in the best-performing LmDiv model trained for our BM25 run (over TREC 2009-2012 topics). As we have applied a 5-fold CV during training, for each year's query set, a feature's weight is the average weight over those obtained from the LmDiv models built for five different training folds. We also provide the overall average weight of a feature (over these 4 query sets) in the last column of the table.

Table 3.7 reveals that all features are likely to contribute positively in most of the cases. Specifically, the last two post-retrieval features, VScoreAvg and VScoreFirst, are consistently weighted higher, implying that they are more useful for the LmDiv models. Having said that, we also observe that the other features are found to be useful, and even the features that have negative weights on the average (maxSCQ and NQC) have positive weights for certain query sets. For this reason (and due to the fact that our models indeed have a moderate number of features, in comparison to the LTR models, which include hundreds of features), we prefer to keep all of them in

Table 3.7: The weights of QPP features in the LmDiv framework for the BM25 runs (over TREC 2009-2012 topic sets).

QPP Feature	2009	2010	2011	2012	Average
maxSCQ	-0.046	0.040	-0.062	0.003	-0.016
sigma	0.037	0.315	-0.083	0.173	0.111
WIG	0.362	0.414	0.145	-0.213	0.177
NQC	-0.116	-0.064	-0.003	0.062	-0.030
ScoreAvg	-0.031	-0.062	-0.014	0.200	0.023
ScoreDev	0.290	-0.046	0.033	0.315	0.148
ScoreRatio	-0.045	0.104	-0.056	0.083	0.021
VScoreAvg	0.373	0.274	0.100	0.307	0.263
VScoreFirst	0.440	0.401	0.213	0.260	0.328

our models. Note that this choice does not incur a significant efficiency overhead for diversification during the run-time, as we discuss next.

While our results up to this point demonstrate the superiority of the LmDiv framework in terms of diversification effectiveness, as we aim to propose approaches that are applicable in practical scenarios, we also provide a comparison of the algorithmic complexity of LmDiv and the baseline xQuAD approach.

To begin with, as discussed in Section 3.3.1, xQuAD constructs the final top-k ranking iteratively, by selecting the document that maximizes Equation (3.1), which is computed for each document in the candidate set D in each iteration. Hence, the complexity of xQuAD is $\mathcal{O}(Nk)$, where $N = |D|$.

For the LmDiv framework, during diversification, the first step is preparing the feature vectors over the candidate set, which requires computing the QPP features. At this point, we would like to emphasize that xQuAD (like many other unsupervised methods, such as IA-Select, PM2, etc.) computes each candidate document's relevance score for each aspect (i.e., $s(d_j, a_i)$), hence the overhead of computing the post-retrieval QPPs (in comparison to xQuAD) is very low, and essentially amounts to obtaining the top-n ranking $D^n_{a_i}$ of these documents for each aspect (note that, as

$n \approx k$ in our setup, we use k for simplicity in this analysis). Since k is smaller than N (by an order of magnitude, in some cases), we can obtain the rankings $D^k_{a_i}$ efficiently, by extracting the top-k documents of an aspect in $\mathcal{O}(k \log N)$ time using a size-*N* max-heap. Thus, the complexity of generating aspect rankings (to compute QPP features) for $|A|$ aspects is $\mathcal{O}(|A|k \log N)$. During the final score computation, LmDiv takes the feature vectors generated for each document and aspect pair as the input, implying $\mathcal{O}(N|A|)$ time complexity. Thus, the overall complexity of LmDiv is $\mathcal{O}(N|A|)$+$\mathcal{O}(|A|k \log N)$; which is comparable to and even better than that of xQuAD, for practical values of N (between 50 and 1000), $|A|$ (at most 10) and k (at most 20), as also employed in the literature. In our experiments, we also observed that the run-time processing efficiency of LmDiv is better than that of xQuAD.

To summarize, our findings in this section indicate that learning a model for obtaining aspect importance values and scoring documents, simultaneously, is the most effective approach (especially, in terms of the ERR-IA and α-nDCG metrics addressing both diversity and novelty) for applying supervised learning in explicit search result diversification. Furthermore, as reported in other scenarios [62], a deeper neural network (i.e., with a larger number of hidden layers) is likely to yield a better performance than a shallow one. Overall, we conclude that our best-performing framework, LmDiv, is both effective and efficient for diversification.

3.6.2 Diversification Performance of the LmDiv Framework for the TREC Runs

To demonstrate the robustness of the best-performing framework, namely, LmDiv, we conduct additional experiments. To this end, we use the submitted runs that exhibit the highest α-nDCG@20 score in the ad hoc retrieval track of TREC between 2009 and 2012. Note that since these runs are not diversified, they can safely serve as the candidate sets (generated with various ranking methods beyond BM25), following the practice in [92, 93].

In Table 3.8, we present the effectiveness of the LmDiv framework against the baselines by averaging the metric scores over the four different TREC runs (listed in Section 3.5). We observe that both LmDiv versions outperform all the baselines for the ERR-IA and α-nDCG metrics, while their performance is inferior to the strongest

Table 3.8: Diversification performance of the LmDiv framework for the TREC Runs (averaged over the four best-performing runs corresponding to TREC submissions between 2009-2012). The superscripts with (†), (∗), (‡) denote a statistically significant difference from NonDiv, xQuAD, xQuAD_{SR} at 0.05 level, respectively.

	TREC Runs				
Method	ERR-IA@20	α-nDCG@20	P-IA@20	ST-Recall@20	MAP-IA@20
NonDiv	0.3380	0.4452	0.1868	0.6564	0.0416
xQuAD	0.3709	0.4829	0.2091	0.6886	0.0511
xQuAD_{SR}	0.3851	0.4936	**0.2137**	0.6879	**0.0528**
LmDiv-S	**0.3950**†,*(2.6%)	0.5018†,*(1.7%)	0.2126†(-0.5%)	0.6850†(-0.4%)	0.0522†(-1.1%)
LmDiv-D	0.3913†,*(1.6%)	**0.5041**†,*(2.1%)	0.2040†(-4.5%)	**0.7086**†,‡(3.0%)	0.0517†(-2.1%)

baseline, xQuAD_{SR}, for P-IA and MAP-IA (but again, better than NonDiv and/or xQuAD). In this case, there is no clear winner between the two LmDiv versions, yet LmDiv-Deep yields the highest relative improvements for the α-nDCG and ST-Recall metrics.

Next, as in the previous section, we focus on the α-nDCG metric at different rank cut-offs, and present results for each of the selected TREC runs in Tables 3.9-3.12.

Table 3.9 presents the diversification performance of LmDiv for the best-performing TREC 2009 run (*Ucdsiftinter*). In this case, LmDiv-Deep yields the highest scores for the top-10 and top-20 results, and again provides gains that reach up to 37% (0.2061 $\rightarrow$ 0.2816), 12% (0.2519 $\rightarrow$ 0.2816), 6.9% (0.2888 $\rightarrow$ 0.3086) over NonDiv, xQuAD and xQuAD_{SR}, respectively.

In Table 3.10, we report the results for *uogTrB67* from TREC 2010. The findings in this case are slightly different in that LmDiv-Shallow outperforms its Deep version. In particular, LmDiv-Shallow provides a relative improvement of 8.1%, 2.5% and 1.9% over xQuAD_{SR}, for cut-off values 2, 10 and 20, respectively. Nevertheless, both LmDiv approaches still outperform all the baselines for almost all cut-off values.

In Tables 3.11 and 3.12, we present our findings for the TREC 2011 and 2012 runs

Table 3.9: Diversification performance of the LmDiv framework for the best-performing TREC 2009 run, *Ucdsiftinter*. The superscripts with (†) denote a statistically significant difference from NonDiv at 0.05 level. For the LmDiv variants, % gains w.r.t. $xQuAD_{SR}$ are in parentheses.

	TREC 2009 Run (Ucdsiftinter)		
Method	α-nDCG@2	α-nDCG@10	α-nDCG@20
NonDiv	0.2061	0.2534	0.2802
xQuAD	0.2519	0.2810	0.3010
$xQuAD_{SR}$	0.2833	0.2888	0.3116
LmDiv-S	**0.2996**†(5.7%)	0.2983†(3.3%)	0.3209†(3.0%)
LmDiv-D	0.2816†(-0.6%)	**0.3086**†(6.9%)	**0.3265**†(4.8%)

Table 3.10: Diversification performance of the LmDiv framework for the best-performing TREC 2010 run, *uogTrB67*. The superscripts with (†) denote a statistically significant difference from NonDiv at 0.05 level. For the LmDiv variants, % gains w.r.t. $xQuAD_{SR}$ are in parentheses.

	TREC 2010 Run (uogTrB67)		
Method	α-nDCG@2	α-nDCG@10	α-nDCG@20
NonDiv	0.3378	0.3717	0.4178
xQuAD	0.3400	0.4146	0.4584
$xQuAD_{SR}$	0.3682	0.4325	0.4734
LmDiv-S	**0.3979**(8.1%)	**0.4434**†(2.5%)	**0.4822**†(1.9%)
LmDiv-D	0.3714(0.9%)	0.4359†(0.8%)	0.4754†(0.4%)

(with ids *Srchvrs11b* and *Qutparabline*, respectively). LmDiv-Deep is again the best-performing approach for diversification of both of these runs. According to Table 3.11, LmDiv-Deep provides relative gains of 11.6%, 4.3%, 4.1% over $xQuAD_{SR}$, for cut-off values 2, 10 and 20, respectively. Table 3.12 also reveals improvements, reaching up to 1.42% over $xQuAD_{SR}$.

Table 3.11: Diversification performance of the LmDiv framework for the best-performing TREC 2011 run, *Srchvrs11b*. The superscripts with (†) and (∗) denote a statistically significant difference from NonDiv and xQuAD at 0.05 level, respectively. For the LmDiv variants, % gains w.r.t. xQuAD_{SR} are presented in parentheses.

	TREC 2011 Run (Srchvrs11b)		
Method	α-nDCG@2	α-nDCG@10	α-nDCG@20
NonDiv	0.4356	0.5312	0.5546
xQuAD	0.4630	0.5510	0.5747
xQuAD_{SR}	0.4657	0.5606	0.5872
LmDiv-S	0.5101*(9.5%)	0.5715(1.9%)	0.6033*(2.7%)
LmDiv-D	**0.5198**(11.6%)	**0.5848**†(4.3%)	**0.6111**†,*(4.1%)

Table 3.12: Diversification performance of the LmDiv framework for the best-performing TREC 2012 run, *Qutparabline*. The superscripts with (†) denote a statistically significant difference from NonDiv at 0.05 level. For the LmDiv variants, % gains w.r.t. xQuAD_{SR} are in parentheses.

	TREC 2012 Run (Qutparabline)		
Method	α-nDCG@2	α-nDCG@10	α-nDCG@20
NonDiv	0.4013	0.4943	0.5269
xQuAD	0.5011	0.5710	0.5963
xQuAD_{SR}	0.4971	0.5680	0.6014
LmDiv-S	0.5021†(1.0%)	0.5744†(1.1%)	0.6001†(-0.2%)
LmDiv-D	**0.5024**†(1.1%)	**0.5761**†(1.4%)	**0.6022**†(0.1%)

Our findings in this section are important. In earlier works [92, 93], it has been shown that providing an impressive diversification performance is more challenging when the initial retrieval results (i.e., NonDiv) are produced via sophisticated methods. Contrary to this, here we demonstrate that the LmDiv framework considerably improves the diversification performance (especially for the ERR-IA and α-nDCG

metrics) over strong baselines (improvements being statistically significant for NonDiv and/or xQuAD) and using the best-performing ad hoc runs from different years and research groups. Therefore, the findings in this section justify our goal of employing supervised learning in explicit diversification, and further reveal that a particular framework, LmDiv, is the most effective and robust approach to achieve this goal.

3.7 Conclusions

In this chapter, we sought an answer to the following key question: How can we exploit supervised learning methods to improve the effectiveness of explicit search result diversification? We identified three directions to achieve this goal, leading to three frameworks leveraging supervised learning with different features and goals. In the LTRDiv framework, we formulated the diversification problem as that of learning a ranking model, which is based on the aggregated relevance of each document to all aspects of a given query, and used well-known LTR algorithms, such as SVMRank and Random Forests. In our second framework, AspectRanker, we addressed a critical sub-task for result diversification and trained models (using various query performance predictors as features) to predict the importance of each query aspect. Then, these predicted values are exploited by a traditional (unsupervised) explicit diversification method. Finally, in the LmDiv framework, we adapted the LambdaMerge approach [1] for the supervised merging of rankings per query aspect. Our exhaustive experiments over the standard TREC diversification topic sets (between 2009-2012) and using initial retrieval results generated by our group and other TREC participants justified the necessity and success of using supervised learning in this context. All three frameworks provided impressive improvements over the baselines.

CHAPTER 4

DIVERSIFICATION BASED STATIC INDEX PRUNING

In this chapter, we turn our attention to static index pruning, which aims to remove redundant parts of an index, and examine static index pruning from the diversification perspective. We focus on that how static index pruning affects diversification performance and how we can improve diversification effectiveness while pruning.

This chapter is structured as follows. In the next section, we provide the motivation for our work. In Section 4.2, we review the earlier works in the context of static index pruning. Then, we introduce our diversification-based static index pruning strategies. The experimental setup and results are presented in Section 4.4 and Section 4.5, respectively. We conclude this chapter in Section 4.6.

4.1 Introduction

IR deals with returning a ranked document list with respect to the relevance of documents from within large collections aiming to fulfill the information need expressed by the user as a query [2]. The challenge of finding such a list across massive amounts of documents is mainly accomplished by using an inverted index in IR systems. An inverted index is a data structure that correlates index terms and documents with their occurrence. An example of an inverted index is illustrated in Figure 4.1.

Index terms are referred as dictionary (a.k.a lexicon or vocabulary). Each term has a list, called a postings list, which comprises of a *document id* the term has appeared in, and a *payload* that contains information about term occurrences such as term frequency, position of every occurrence of the term in the document, etc. [95].

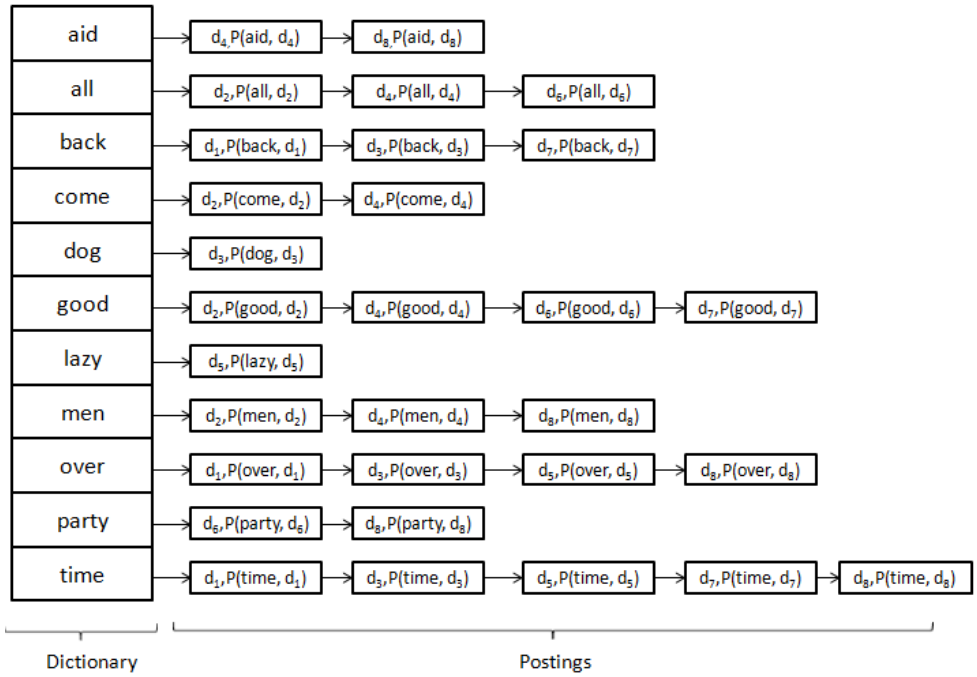


Figure 4.1: Inverted Index

For small collections, memory may hold the entire inverted index, but for large collections (such as WWW), memory would be inadequate to fit it all, so inverted index resides on disk [96]. The more the size of inverted index increases, the more storage overhead increases. Since IR systems have to respond hundreds of millions of search queries per day, they need to be efficient to keep low response times. The efficiency of the retrieval system is attached to query processing performance. The amount of disk accesses while processing a query (i.e., getting postings list from the disk for each term) is vital affecting the query response time directly.

As the growth in the number of documents elongates the postings list to be processed, even it can reach hundreds of MBs or even GBs for common terms, it increases query response time readily. Hence, how to process the index effectively and the decision of which data to keep in the index is a challenging problem in IR. An approach in the literature to tackle this challenge is "pruning". Pruning is a novel technique that aims to reduce the number of documents processed, while preserving the quality of retrieval performance (which means to keep the original top-k ranking as much as possible).

Pruning methods can be applied dynamically, i.e., during query processing, or statically. While processing a query, firstly, the postings list of each query term is retrieved from the index. Then, a cumulative score is obtained by adding the similarity score of each document in the postings lists, and documents contributing to reach the highest cumulative score are returned as the search result list. Dynamic pruning determines which documents (or postings) will be included in the accumulated score and when ranking process ends [97]. The decline in the number of postings reduces query response time by implying the shortened navigation over postings.

Static index pruning is performed in advance of query processing. It is independent from queries. That means, it can be executed off-line [98]. Static index pruning removes less important postings, which are decided according to a relevance criterion (i.e. tf-idf or BM25), from the index. It not only reduces the query response time, but also saves disk storage space. These two approaches (namely, static index pruning and dynamic pruning) can be used separately or together. For instance, in addition the use of the pruned index, retrieval performance would be speeded up by dynamic pruning at the time of query processing.

In this study, we have two main contributions:

1. We analyze the impact of static index pruning on the result diversification measured by various diversity-aware metrics. While doing so, we employ several well-known static index pruning techniques, namely, Popularity-based Pruning (PP), Access-based Term-Centric Pruning (aTCP), Access-based Document-Centric Pruning (aDCP) algorithms.

2. We propose two new diversity-aware static index pruning strategies to preserve diversification effectiveness over pruned indexes.

4.2 Related Work

Static index pruning strategies vary depending on whether pruning goes over terms or documents, and are categorized as term-centric and document-centric.

A pioneering study in static index pruning was represented by Carmel et al. [97]. In

their first approach, called *uniform*, they proposed a fixed threshold to all terms in the vocabulary. All entries in postings lists are sorted by SMART scoring function, those below the threshold are thrown out of the index. Their next approach, which is more successful, individually defines a threshold value for each term. The documents in the postings list of a term are sorted in descending order, then the threshold value of the term is computed as $\epsilon * z_t$ where z_t is the k^{th} highest score of term t and ϵ is a parameter to control pruning level. The postings under the threshold are accepted as less important documents and removed from the index. In this study, we refer to the latter algorithm as *term-centric pruning (TCP)*. The pseudocode of the TCP algorithm is given in Algorithm 2 (based on the version in [99]).

Algorithm 2: Term-Centric Pruning (TCP)

Input : I : inverted index
k : indicates which the highest score to be chosen
ϵ : pruning level

Output: I_p : pruned index

```
for each term t in I do
    get the postings list P_t from I
    if |P_t| > k then
        for each document d in P_t do
            calculate a document score, s(t, d)
        end
        sort P_t in descending order w.r.t. s
        z_t ← k^th highest score
        τ_t ← ϵ × z_t
        for each document d in P_t do
            if s(t, d) <= τ_t then
                remove document d from P_t
            end
        end
    end
    set (t, P_t) in I_p
end
```

Chen and Lee [100] adapted pruning as a convex integer program providing a theoretical foundation. They re-studied uniform pruning with KL divergence, and obtained impressive mean average precision (MAP) results, but some technical issues were left unsettled. Their follow-up study [101] explores different divergence measures to overcome these issues.

Moura et al. [102] introduced a new pruning method that takes into account the positional index in addition to the typical (frequency) index to handle conjunctive and phrase queries. In [103], they improved their method that enables the pruning process to be performed simultaneously with index construction or updating.

Blanco [104] proposed to eliminate stop words with their postings lists from the index to reduce the index size. They employed three techniques to identify stopwords: inverse document frequency (IDF), residual inverse document frequency (RIDF) and term discrimination model. Their next study [98] introduced a novel pruning strategy based on probability ranking principle (PRP) [105, 106] which is widely used for document retrieval in IR. They considered every term in the dictionary as a single-term query and decided whether the document would remain in the index depending on three distinct probabilities (i.e., document prior, query likelihood, probability of a term being non-relevant).

In contrast to term-centric strategies that discard some postings of terms, Büttcher and Clarke [107] introduced *document-centric pruning (DCP)* strategy that removes some terms from documents to reduce the size of documents as it would also reduce the size of the index. Inspired by [108], they assessed each term's contribution to the document's content using Kullback-Leibler divergence (KLD), and then stored top-scoring terms in the document and dispose of others. They presented two instantiations of the method: (1) they stored top-k (fixed k) terms of each document at the index; (2) the number of terms to be pruned in a document are decided according to the count of distinct terms in the document. The intuition behind this approach is that larger documents are likely to have a wide variety of topics implying more possible query terms. The latter yields better performance. In [99], they applied DCP to all terms in the index instead of only the 10^6 most frequent terms as in [107], and counted all terms rather than unique terms while computing document size. The steps of this

Algorithm 3: Document-Centric Pruning (DCP)

Input : D : document set in the collection

ϵ : pruning level

Output: I_p : pruned index

```
for each document d in D do
    for each term t in d do
        calculate s(d, t) with KLD
    end
    sort scores in descending order
    remove the last |d| × ε terms from d
    set d in I_p
end
```

algorithm are shown in Algorithm 3 (based on the version in [99]).

Zheng and Cox [109] represented another document-centric pruning strategy that determines documents to be pruned by assessing the entropy of terms in the document. They argued that as the number of distinctive terms (i.e., scores with low entropy) in a document increases, this document could be considered as more important than the others in the collection. They entirely removed all postings regarding the documents which are less important. But, due to the improvement in performance over different collections is not stable, they suggested a hybrid approach which 10% of the index is pruned by entropy-based method and the rest of amount by Carmel's method [97]. Vishwakarma et al. [110] also performed a pruning in which, similar to [109], all postings of the document are removed from the index, using a different scoring function based on tf-idf.

Nguyen [111] discussed pruning in terms of postings by combining term- and document-centric approach. Each posting re-defined as <term, document, term frequency> passes through a ranking function that considers both informativeness of the term and importance of document among the collection, and some of them are eliminated from the index according to their scores until desired pruning level is reached.

Algorithm 4: Access-based Term-Centric Pruning (aTCP)

Input : I : inverted index
AC_D: a list indicating an access count value for each document in D
ϵ : pruning level

Output: I_p : pruned index

```
for each term t in I do
    get the postings list P_t from I
    sort documents in P_t w.r.t. access counts, AC_D, in descending order
    remove the last ε × |P_t| from P_t
    set (t, P_t) in I_p
end
```

The experiments demonstrated that posting-based approach is better at low pruning levels, but at higher levels does not outperform the performance of DCP.

Garcia [112] attempted to employ a query log of a search engine for static index pruning. Instead of using the relevance score of a document as the scoring function, he utilized the access count of the document, which is defined as the number of times the document occurs in the top-k search results of any query. k can be 10, 100, and 1000. This work achieves 75% improvement in query response time, yet damages accuracy up to %22 in MAP since for each term a constant number of postings with the top access counts are kept.

Altingovde et al. [99] enhanced this approach, named *access-based term centric pruning (aTCP)*, by keeping a fraction (ϵ) of postings for every term in dictionary. ϵ represents the pruning level. This algorithm is outlined in Algorithm 4. In [99], a novel strategy called *access-based document-centric pruning (aDCP)* is proposed, as well. Access count information is employed from document-centric pruning perspective in contrast to term-centric where postings from each term list are pruned. They discarded documents with the lowest access count values from the index until a condition bounded to collection length is reached. The collection length, $|C|$, is obtained by adding the number of distinct terms in every document of the collection. A sketch of this algorithm is given in Algorithm 5.

Algorithm 5: Access-based Document-Centric Pruning (aDCP)

Input : I : inverted index
$|C|$: collection length
AC_D: a list indicating an access count value for each document in D
ϵ : pruning level

```
1 sort AC_D w.r.t. access counts in descending order
2 totalPrunedPostings ← 0
3 while totalPrunedPostings < ε × |C| do
4     remove the document d with the lowest access count in AC_D
5     totalPrunedPostings ← totalPrunedPostings + |d|
6 end
```

The usage of popularity of terms for pruning and caching [99, 113, 114, 115] is a straightforward but powerful method. In this pruning method, so-called *Popularity-based Pruning (PP)*, pruning is performed based on a score assigned for each term in a query log. This score is computed by dividing term frequency to document frequency. Term frequency is the number of queries containing the term in the query log, in other words, it expresses the popularity of the term. Document frequency represents the number of documents the term appears in the collection. They are inversely proportional since the terms with high document frequency have long postings lists that means they physically occupy considerable space in the disk. Terms are sorted according to their scores, and the lowest ones, i.e. the most unpopular terms, are pruned from the index. The pseudocode of this algorithm is shown in Algorithm 6 (based on the version in [99]).

Altingovde et al. [99] introduced the concept of query view for the static index pruning, and incorporated it into the five different methods mentioned above. Query view of a document is the re-expression of the document with all the words contained in the queries which document appears in the top-k search results. They achieved significant improvement at preserving the top-ranked results intact in comparison to the pure pruning methods.

- Term-Centric Pruning with Query Views (TCP-QV): In TCP, a posting in the

Algorithm 6: Popularity-based Pruning (PP)

Input : I : inverted index
Pop: a list indicating popularity of each term in I
ϵ : pruning level

Output: I_p : pruned index

```
for each term t in I do
    get the postings list P_t from I
    calculate s(t) ← Pop[t]/|P_t|
    isNotPruned_t ← 0
end
sort terms w.r.t. scores s(t) in descending order
totalRemainedPostings ← 0
while totalRemainedPostings < ε × |I| do
    fetch the term t with the highest score s(t)
    isNotPruned_t ← 1
    totalRemainedPostings ← totalRemainedPostings + |P_t|
end
for each term t in I do
    if isNotPruned_t then
        set (t, P_t) in I_p
    end
end
```

list of term t is discarded from the index if its score is under the threshold, τ_t. As for TCP-QV, before discarding the posting, it's checked whether the term is part of the query view of the document referring to the posting. If not, it's pruned, otherwise stored in the index.

- Document-Centric Pruning with Query Views (DCP-QV): Query views are attached to the DCP algorithm as another key to be used in sorting. All terms in a document are sorted first by whether the term is in query view of the document, QV_d, and then by their scores. Next, the last $\epsilon \times P_t$ terms are omitted.

- Access-based Term-Centric Pruning with Query Views (aTCP-QV): In aTCP-

QV, while sorting postings, we have two keys to find the postings to be pruned. The primary sort key is whether or not the term is part of the query view of the document indicating the posting. The secondary key is the access count of the document. While pruning, the last $\epsilon \times P_t$ postings are removed.

- Access-based Document-Centric Pruning (aDCP) with Query Views (aDCP-QV): aDCP algorithm eliminates documents starting from with the lowest access counts until the pruning level is reached. But this time, instead of eliminating the document directly, all terms in the document are examined according to their appearance in the query view, QV_d. If a term is not the member of QV_d, the term is discarded from the document, otherwise preserved.

- Popularity-based Pruning (PP) strategy with Query Views (PP-QV): This procedure works as follows: if the size of the pruned index is not sufficient to include all query views, the index is pruned according to the highness of the popularity score. If there is still some space available after insertion of the query views of all terms, the query view of the term is substituted with the corresponding complete postings list starting from the highest value in terms of popularity.

To the best of our knowledge, pruning from the diversification perspective has not yet been addressed enough. The only work that performs pruning based on diversification is Pehlivan et al. [116]. While pruning, they considered to preserve both retrieval performance and diversity. To this end, the next posting to be kept is selected in a greedy manner to optimize an evaluation metric, i.e., DCG. They used two temporal collections in which documents and queries are correlated with temporal aspects. The diversification on their work is time-aware, in other words, they diversify documents based on their time interval. Conversely, we focus on topical diversification rather than temporal. We identify documents over their topical content, and provide diversity among them in terms of topical variety while pruning documents, and evaluate the ranking by diversity-aware metrics such as α-NDCG [20], Precision-IA [4] unlike [116] that used relevance metrics (i.e., NDCG and MAP) for evaluation.

4.3 Diversity-Aware Static Index Pruning Approaches

Pruning strategies are mainly based on relevance scores (i.e. BM25, tf-idf) to identify documents to be pruned. We envision that if topically diverse documents are eliminated from the index, then pruning may lead to the disappearance or favoring of an aspect of a topic, which implies a drop in diversification effectiveness and inducing accessibility bias [117, 118]. Furthermore, pruning diverse documents affects the performance of subsequent diversification step. For instance, "java" is an ambiguous query. There might be documents in the collection related to an island in Indonesia, a type of coffee, or a computer programming language. By the way, these aspects of the query "java" is still not precisely specified to understand the user information need. The "computer programming language" aspect has different sub-aspects such as *how to build java*, *ides to develop java codes*, *how to learn java coding*. If any pruning algorithm removes all the documents related to an aspect from the index, say, island which appears in fewer documents compared to other aspects, the existence of this aspect in the collection would end and cause a negative effect for further tasks. To remedy this problem, we propose two new pruning approaches that recognize topical variety of documents while pruning.

Note that we have no queries and/or information about the relationships of the query aspects and documents to be exploited during pruning unlike the diversification process in the previous chapters since static index pruning is performed prior to query execution.

4.3.1 Access-based Term-Centric Diversity-Aware Pruning (aTCP-Div)

As our first approach towards developing diversity-aware pruning strategies, we adapt the aTCP strategy that pays attention to topics of documents over the postings lists of terms for the purpose of diversification.

Clustering is one of the widely used unsupervised learning techniques for the discovery of topics from documents. It aims to generate clusters in a way that documents in the same cluster are similar to each other according to documents in other clusters and each cluster refers to a different topic [119]. Therefore, we utilize clustering to detect

Algorithm 7: Access-based Term-Centric Clustering-Based Diversity-Aware Pruning (aTCP-Div-Clust)

Input : I : inverted index
AC_D: a list indicating an access count value for each document in D
ϵ : pruning level
map: the document-cluster mapping

```
for each term t in I do
    get the postings list P_t from I
    sort documents in P_t w.r.t. access counts, AC_D, in descending order
    create buckets based on map
    for each bucket b do
        NumPrunedPostings ← 0
        while NumPrunedPostings < ε × |b| do
            remove the document d with the lowest access count
            NumPrunedPostings ← NumPrunedPostings + 1
        end
    end
end
```

the topical diversity of documents as we have only vocabulary terms and document content for accomplishing the diversification task.

We employ clustering over the entire collection and obtain a general document-cluster mapping. We use this clustering structure to capture the semantics of documents on the term's postings list. We compose a bucket for each cluster observed in the term's postings list and sort postings in each bucket individually in decreasing order according to their access count. Then, we prune each bucket proportional to the pruning level. Algorithm 7 shows this strategy.

Query expansion is a well-known technique to improve retrieval effectiveness of a search system by minimizing vocabulary mismatch problem that occurs when index terms and query terms are not based on the same set of words (e.g. "buy" and "purchase") [120]. It expands the original query by finding out semantically simi-

lar terms. As depicted in Figure 4.1, the terms in an inverted index consist of one word and it's too short to capture accurately term-document relations. To this end, we utilize a query expansion technique by enriching terms in the vocabulary of the index to identify the topical diversity of documents. Finding documents more relevant to the expanded terms that consist of semantically similar words would reveal semantically coherent documents. Similarly, the diverse expanded terms help to find topically diverse documents.

Bouchoucha et al. [121] proposed MMRE method to diversify query expansion terms as a pre-step for diversification of search results. MMRE is the adaption of MMR which is a pioneer implicit diversification method that considers the document relevance to the query and dissimilarity of the documents already selected. As in MMR, MMRE iteratively selects expansion terms with having the highest scores based on the following formula:

$$score_{MMRE}(t_i) = \lambda \Pr(t_i|q) - (1-\lambda) \max_{w' \in S} \Pr(t'|q) \tag{4.1}$$

where t_i is the candidate term, $\Pr(t_i|q)$ is the relevance of t_i to query q, and S is the set of expanded terms already selected.

In [122], a query expansion technique using word embeddings is applied to generate diverse and similar query sets to be used in a broker node for distributed search architecture. We adopt this approach to find similar and diverse words of the terms in the vocabulary using an adaptive version instead of a fixed number for each word. We select terms whose cosine similarity scores are above a threshold as similar terms. The diverse terms are selected depending on whether or not the score of MMRE [121] is higher than the threshold, which is set to 0.5.

While scoring postings of a term, similar or diverse terms of the corresponding term are considered with the term itself. We compute a total score of a posting in the related term by adding up relevance scores of similar or diverse terms. Since access count is a strong driver to identify documents to be pruned, it is also taken into account for the final score of the posting. The score of each posting in the postings list of term t is computed as follows:

Algorithm 8: Access-based Term-Centric Diversity-Based-Word Embeddings Pruning (aTCP-Div-WE)

Input : I : inverted index

AC_D: a list indicating an access count values for each document in D

ϵ : pruning level

exW: a list representing expanded (diverse or similar) words of the terms

```
for each term t in I do
    get the postings list P_t from I
    get the expanded words of t from exW, e_t
    for each document d in P_t do
        s(t, d) ← 0
        for each word w in e_t do
            compute a document score, s(w, d)
            s(t, d) ← s(t, d) + s(w, d)
        end
    end
    sort documents in P_t in descending order w.r.t. s(t, d)
    remove the last ϵ × |P_t| from P_t
end
```

$$Score_{(p,t)} = log(AC[d]) \cdot \Big[s(t,d) + \sum_{w \in ex_W} s(w,d) \Big] \tag{4.2}$$

where $AC[d]$ represents the access count value of the posting corresponding document d. $s(t,d)$ and $s(w,d)$ are the relevance scores of the document d (i.e., BM25) with respect to term t and word w, respectively. ex_W is the set of expanded terms that are similar or diverse words of term t. Once the scoring of documents in the postings list is completed, they are sorted in descending order and pruned starting from the lowest score until the desired pruning level is reached. Algorithm 8 displays this pruning strategy, which we call aTCP-Div-WE.

Algorithm 9: Access-based Document-Centric Diversity-Aware Pruning (aDCP-Div)

Input : I : inverted index
AC_D: a list indicating an access count value for each document in D
ϵ : pruning level
map: a document-cluster mapping

```
create buckets based on map
for each bucket b do
    sort documents in b w.r.t. access counts in descending order using AC_D
    NumPrunedPostings ← 0
    while NumPrunedPostings < ε × |b| do
        remove the document d with the lowest access count from bucket b
        NumPrunedPostings ← NumPrunedPostings + 1
    end
end
```

4.3.2 Access-based Document-Centric Diversity-Aware Pruning (aDCP-Div)

Here, we tailor the aDCP approach, which runs over documents, in the manner of taking into account diversification. We again benefit from clustering for our task of finding diverse documents, i.e., to identify the topical variety of documents. Using a document-cluster mapping, we place each document in the collection to the corresponding cluster-bucket and then sort them in descending order of their access counts. We prune the documents in each bucket entirely from the collection, in the ratio of the pruning level, starting from the document with the lowest access count. Consequently, we keep documents with high access counts and topically diverse. To break the ties among documents with the same access counts, we use document IDs as an intermediate key for sorting. Algorithm 9 shows our pruning strategy, which we call aDCP-Div.

URL is a quite informative and useful source for getting semantic information about the document and employed in different contexts, such as topic classification [123, 124], document annotation [125] as it gives accurate estimates. Hence, we benefit

from URLs of documents to detect topical representation of documents as an alternative clustering instead of that of using document content. We use the second-level domain attribute of URL as the representative for a document, and cluster all documents in the collection based on their URLs (more details are provided in the next section). In a nutshell, we propose two variants of aDCP-Div with employing different document-cluster mappings and refer to them as aDCP-Div-Clust and aDCP-Div-URL hereafter, respectively.

4.4 Experimental Setup

In this study, we used the Category B subset of ClueWeb09 collection. This subset contains 50 million English pages that are crawled from the web in 2009. The dataset is indexed by the Zettair IR system[1]. In the course of indexing, stop-words and numbers are involved, and there is no stemming. The index yields a dictionary containing about 160 million terms and occupies 136 GB of disk storage space (uncompressed).

Our experiments to evaluate the performance of proposed pruning strategies are conducted on this index. We pruned the index at various pruning levels by adjusting the parameter ϵ which ranges from 60% to 90%. We have not experimented with lower pruning levels since, as stated in [99], the effectiveness of pruning algorithms becomes prominent at the high pruning levels where the relevance effectiveness of the pruned index is comparatively lower compared to the unpruned index. We employ query (topic) sets that have been developed for the Diversity Task of the TREC Web Track between 2009 and 2012. Each set includes 50 queries (except for 2010, which has 48) and relevance judgments. The queries are executed in the disjunctive mode on the pruned index and top-1000 results per query are retrieved by using our own retrieval system implementing the traditional BM25 model. We opt for BM25 because it is still the most widely used IR model in practical settings, (e.g., as a ranking feature in Yahoo [91]), and it does not require additional features. Furthermore, we employ BM25 in our proposed aTCP algorithm while computing scores of documents since BM25 is a strong candidate as the scoring function [126]. The parameters of BM25, namely k1 and b, are set experimentally to 1.2 and 0.5, respectively. We also

[1] www.seg.rmit.edu.au/zettair/

follow the practice in [127] where terms that have document frequency, df, greater than $\frac{N}{2}$ are discarded from the index as they generate negative scores in the BM25's idf formula, $log(\frac{N-df+0.5}{df+0.5})$, where N is the total number of documents in the collection. As shown in the studies conducted on ClueWeb09 collection, spam filtering substantially improves the initial retrieval performance. Thus, we use the spam filtering method in [91] where a spam score for each document in ClueWeb09 collection is computed. Based on the score values identified in the Waterloo Spam Ranking[2], we consider documents with a spam score of 60 or less as containing spam and remove these documents during ranking.

Note that while computing the access count of documents, we go through the top-1000 search results of queries, and there might be a lot of documents with the same access counts. To break ties, following the [99], we sort URLs of documents in lexicographical order.

Clustering. We use the document-cluster mapping provided by QKLD-QInit that is the best-performing method in [128] for entire documents in the ClueWeb09 Category B dataset. Dai et al. partitioned the document collection depending on a cluster-based approach for selective search purposes [128]. They employed k-means clustering with the query-driven seeds which are extracted from a query log to be semantically close topics and used a similarity metric based on KLD. It contains 123 clusters.

In URL-based clustering in which we use second-level domain attribute of a URL as a key for the document's topic, we obtain nearly 3 million clusters, from which it is hard to extract meaningful and comprehensible information to be used in pruning. Therefore, we sort the clusters in the order of their sizes and get the largest 999 clusters and merge the rest as a big cluster named "Others". This procedure yielded us 1000 clusters.

Baselines. As a baseline, we essentially use the performance of the unpruned index and interpret our results in a way that how we are close to these results. We also employ aTCP and aDCP algorithms to show improvements of our proposed pruning strategies over them.

[2] plg.uwaterloo.ca/ gvcormac/clueweb09spam/

Evaluation. We report results in terms of the widely used diversification metrics, namely, ERR-IA [19], α-nDCG [20], Precision-IA [4] and ST-recall [36], at the cut-off value of 20. The ndeval software[3] is used to compute the diversification metrics. We use the Student's two-tailed paired t-test (at 95% confidence level) for analyzing statistical significance.

4.5 Experimental Results

Our experiments in this section seek answer to the following research questions:

- RQ1: How do the static index pruning techniques, namely, PP, aTCP, aDCP algorithms, affect diversification effectiveness?
- RQ2: How can a method be aware of the diverse postings and preserve them while pruning to alleviate significant drop of diversification performance?

Since most of the studies related to static index pruning are focus on only the relevance of results, as the answer of the first question, we first provide the diversification performance of PP, aTCP, and aDCP algorithms by measuring with diversity-aware metrics.

Note that, we solely rank documents in descending order employing an effective document ranking approach, i.e. BM25, we do not involve any diversification step such as employing xQuAD, PM2, etc.

Table 4.1 represents the diversification effectiveness of the PP algorithm at different pruning levels. The "Org" tag in the table denotes the diversification performance when there is no pruning over the index. As expected, when the pruning level increases, the gap between the results of Org and PP widens due to an increase in the number of pruned documents. It can be seen that when using 40% of the original index, the results are substantially lower reaching up to 16% drop in the ERR-IA metric. For instance, at 90% pruning, PP yields 0.1278, 0.1814, 0.0783, 0.3112 whereas scores of Org are 0.2629, 0.3568, 0.1558, 0.5705, suggesting a relative decline of

[3] `http://trec.nist.gov/data/web10.html`

Table 4.1: Diversification performance of the PP algorithm (over TREC 2009-2012 topic sets) at different pruning levels. The superscripts with $(*)$ denote a statistically significant difference from Org at 0.05 level. In parentheses, we report the percentage change w.r.t. Org.

Pruning Level	ERR-IA@20	α-nDCG@20	P-IA@20	ST-Recall@20
Org	0.2629	0.3568	0.1558	0.5705
90%	0.1278*(-51.4%)	0.1814*(-49.2%)	0.0783*(-49.7%)	0.3112*(-45.5%)
80%	0.2023*(-23.1%)	0.2807*(-21.3%)	0.1241*(-20.3%)	0.4557*(-20.1%)
70%	0.2162*(-17.8%)	0.3020*(-15.4%)	0.1358*(-12.8%)	0.4941*(-13.4%)
60%	0.2201*(-16.3%)	0.3086*(-13.5%)	0.1353*(-13.2%)	0.5126*(-10.1%)

51.4%, 49.2%, 49.7%, 45.5% for ERR-IA, α-nDCG, P-IA, ST-Recall, respectively. Notice that, the aforementioned performance drops in terms of various diversification metrics are all statistically significant, justifying our motivation to develop diversity-aware pruning methods in this work. We observed that there is a case where the score of the P-IA metric diminishes while the pruning level decreases, i.e. difference between 60% and 70% in the P-IA metric. This might be due to the fact that PP overlooks the documents covering more aspects of the query while favoring the documents attached to terms with high popularity.

Table 4.2 displays the diversification performance of aTCP algorithm with different pruning levels. The trends are similar to those in Table 4.1, as the loss in the diversification performance increases in parallel to the pruning level. Our findings show that aTCP outperforms PP for all pruning levels and metrics except P-IA. aTCP yields scores of 0.2116, 0.2946, 0.3228, 0.3348, while PP can only achieve 0.1814, 0.2807, 0.3020, 0.3086 for α-nDCG. When compared to Org (i.e., the unpruned index), the relative decline is 6.3%, 6.2%, 10.6%, 4.8% at the 60% pruning for ERR-IA, α-nDCG, P-IA, ST-Recall, respectively.

Table 4.3 shows the performance of aDCP pruning algorithm from the point of diversification. We again see that scores of diversification metrics rise when the pruning level declines. We also observe that aDCP achieves the best results overall for all

Table 4.2: Diversification performance of the aTCP algorithm (over TREC 2009-2012 topic sets) at different pruning levels. The superscripts with ($*$) denote a statistically significant difference from Org at 0.05 level. In parentheses, we report the percentage change w.r.t. Org.

Pruning Level	ERR-IA@20	α-nDCG@20	P-IA@20	ST-Recall@20
Org	0.2629	0.3568	0.1558	0.5705
90%	0.1612*(-38.7%)	0.2116*(-40.7%)	0.0781*(-49.9%)	0.3492*(-38.8%)
80%	0.2212*(-15.9%)	0.2946*(-17.4%)	0.1152*(-26.1%)	0.4774*(-16.3%)
70%	0.2393*(-9.0%)	0.3228*(-9.5%)	0.1308*(-16.0%)	0.5184*(-9.1%)
60%	0.2464*(-6.3%)	0.3348*(-6.2%)	0.1393*(-10.6%)	0.5432(-4.8%)

Table 4.3: Diversification performance of the aDCP algorithm (over TREC 2009-2012 topic sets) at different pruning levels. The superscripts with ($*$) denote a statistically significant difference from Org at 0.05 level. In parentheses, we report the percentage change w.r.t. Org.

Pruning Level	ERR-IA@20	α-nDCG@20	P-IA@20	ST-Recall@20
Org	0.2629	0.3568	0.1558	0.5705
90%	0.2165*(-17.6%)	0.2899*(-18.8%)	0.1140*(-26.8%)	0.4619*(-19.0%)
80%	0.2405*(-8.5%)	0.3260*(-8.6%)	0.1360*(-12.7%)	0.5270*(-7.6%)
70%	0.2545(-3.2%)	0.3429*(-3.9%)	0.1439*(-7.6%)	0.5516*(-3.3%)
60%	0.2586(-1.6%)	0.3509(-1.7%)	0.1488*(-4.5%)	0.5651(-0.9%)

metrics, that is to say, closing the gap with Org. While aTCP yields 38.7%, 40.7%, 49.9%, 38.8% drop at the 90% pruning, aDCP yields 17.6%, 18.8%, 26.8%, 19.0% drop in ERR-IA, α-nDCG, P-IA, ST-Recall metrics, respectively.

To answer our second question, Table 4.4 compares the diversification effectiveness of three cases: i) when there is no pruning, ii) pruning based on aTCP, and iii) pruning based on our proposed diversity-aware aTCP algorithms. For aTCP-Div-Clust, we also employ the mapping generated by URL-based clustering for document-cluster

Table 4.4: Diversification performance of the diversity-aware aTCP algorithm (over TREC 2009-2012 topic sets) at different pruning levels. The superscript with (†) denotes a statistically significant difference from aTCP at 0.05 level. In parentheses, we report the percentage change w.r.t. aTCP method.

Pruning Level	Method	ERR-IA@20	α-nDCG@20	P-IA@20	ST-Recall@20
Org	-	0.2629	0.3568	0.1558	0.5705
90%	aTCP	0.1612	0.2116	0.0781	0.3492
	aTCP-Div-Clust	0.1823†(13.1%)	0.2373†(12.1%)	0.0886(13.4%)	0.3919†(12.2%)
	aTCP-Div-WE	**0.2078**†(29.0%)	**0.2721**†(28.6%)	**0.1149**†(47.0%)	**0.4362**†(24.9%)
80%	aTCP	0.2212	0.2946	0.1152	0.4774
	aTCP-Div-Clust	0.2254(1.9%)	0.2961(0.5%)	0.1147(-0.4%)	0.4784(0.2%)
	aTCP-Div-WE	**0.2331**(5.4%)	**0.3100**(5.2%)	**0.1286**(11.7%)	**0.5044**(5.7%)
70%	aTCP	0.2393	0.3228	0.1308	0.5184
	aTCP-Div-Clust	0.2355(-1.6%)	0.3187(-1.3%)	0.1315(0.5%)	0.5176(-0.2%)
	aTCP-Div-WE	**0.2539**(6.1%)	**0.3375**(4.5%)	**0.1358**(3.8%)	**0.5411**(4.4%)
60%	aTCP	0.2464	0.3348	0.1393	0.5432
	aTCP-Div-Clust	0.2519(2.2%)	0.3386(1.1%)	0.1423(2.1%)	0.5390(-0.8%)
	aTCP-Div-WE	**0.2640**†(7.2%)	**0.3519**†(5.1%)	**0.1455**(4.4%)	**0.5492**(1.1%)

mapping in Algorithm 7 instead of QKLD-QInit, but we report the results of the latter one as yielding better results. For our second pruning term-centric strategy, namely aTCP-Div-WE, we give the results that employ diverse terms as yielding better performance than similar ones. The proposed diversity-aware aTCP algorithms provide a notable improvement over aTCP for all metrics yielding relative improvements reaching up 47.0% (0.0781 $\rightarrow$ 0.1149), 11.7% (0.1152 $\rightarrow$ 0.1286), 6.1% (0.2393 $\rightarrow$ 0.2539), 7.2% (0.2464 $\rightarrow$ 0.2640) at the 90%, 80%, 70%, 60% pruning, respectively. For instance, at the 90% pruning, aTCP-Div-WE yields relative improvements of 29.0% (0.1612 $\rightarrow$ 0.2078), 28.6% (0.2116 $\rightarrow$ 0.2721), 47.0% (0.0781 $\rightarrow$ 0.1149), 24.9% (0.3492 $\rightarrow$ 0.4362). We compare the diversification performance of aTCP-Div-WE, when more than half of the index (i.e., 60%) is pruned, to the index with no pruning, the relative decline becomes 1.4% (0.3568 vs. 0.3519), 6.6% (0.1558 vs.

Table 4.5: Diversification performance of the diversity-aware aDCP algorithm (over TREC 2009-2012 topic sets) at different pruning levels. The superscript with (†) denotes a statistically significant difference from aDCP at 0.05 level. In parentheses, we report the percentage change w.r.t. aDCP method.

Pruning Level	Method	ERR-IA@20	α-nDCG@20	P-IA@20	ST-Recall@20
Org	-	0.2629	0.3568	0.1558	0.5705
90%	aDCP	0.2165	0.2899	0.1140	0.4619
	aDCP-Div-Clust	0.2172(0.3%)	0.2909(0.4%)	**0.1170**†(2.6%)	0.4659(0.9%))
	aDCP-Div-URL	**0.2207**(1.9%)	**0.2957**(2.0%)	0.1163†(2.0%)	**0.4735**†(2.5%)
80%	aDCP	0.2405	0.3260	0.1360	0.5270
	aDCP-Div-Clust	0.2377(-1.2%)	0.3230(-0.9%)	0.1347(-1.0%)	**0.5279**(0.2%)
	aDCP-Div-URL	**0.2412**(0.3%)	**0.3263**(0.1%)	**0.1367**(0.5%)	0.5254(-0.3%)
70%	aDCP	**0.2545**	**0.3429**	0.1439	**0.5516**
	aDCP-Div-Clust	0.2516(-1.1%)	0.3404(-0.7%)	0.1436(-0.2%)	0.5499(-0.3%)
	aDCP-Div-URL	0.2540(-0.2%)	0.3428(0.0%)	**0.1443**(0.2%)	0.5508(-0.1%)
60%	aDCP	0.2586	0.3509	0.1488	0.5651
	aDCP-Div-Clust	**0.2589**(0.1%)	**0.3515**(0.2%)	**0.1500**(0.8%)	0.5669(0.3%)
	aDCP-Div-URL	0.2574(-0.5%)	0.3505(-0.1%)	0.1491(0.2%)	**0.5685**(0.6%)

0.1455), 3.7% (0.5705 vs. 0.5492), while it is 6.3%, 6.2%, 10.6%, 4.8% for aTCP, for α-nDCG, P-IA, ST-Recall, respectively. Lastly, aTCP-Div-WE outperforms aTCP-Div-Clust for most of the metrics.

Table 4.5 shows the diversification performance of our diversity-aware aDCP pruning strategies versus aDCP. For most of the cases, the proposed aDCP strategies, namely aDCP-Div-Clust and aDCP-Div-URL, outperforms aDCP yielding relative improvements up to 1.9%, 2.0%, 2.6%, 2.5% for ERR-IA, α-nDCG, P-IA, ST-Recall, respectively. The improvements are moderate compared to aTCP-Div since aDCP is the best performing method with respect to Tables 4.1- 4.3 and a strong baseline to beat. When we prune 60% of the index with aDCP-Div-URL, the relative decline in performance versus the total index becomes 2.1% (0.2629 vs. 0.2574), 1.8% (0.3568 vs. 0.3505), 4.3% (0.1558 vs. 0.1491), 0.4% (0.5705 vs. 0.5685) for ERR-IA, α-nDCG, P-IA, ST-Recall, respectively.

Overall, our experiments confirm a positive answer to our second research, diversity-aware pruning strategies are superior to the original pruning approaches, i.e., PP, aTCP and aDCP.

4.6 Conclusions

In this chapter, we analyze the impact of the state-of-the-art pruning strategies from the diversification point of view and reveal how much the pruning strategies damage the diversification performance measured by diversity-aware metrics. We propose simple yet effective term-centric and document-centric pruning strategies that take into account topically diverse documents with the help of clustering and word embeddings. Our experiments show that the proposed methods yield substantial improvements over existing approaches in terms of diversity-aware metrics.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

Search result diversification approaches aim to satisfy more users by promoting documents with diverse content to the top-ranked positions. In this thesis, we propose advanced strategies to improve the diversification effectiveness of search results. More specifically, first, we define a new search result diversification problem that is different from the flat (topical) and hierarchical diversification to fulfill the typical requirements of a search engine in the education domain. To this end, we propose a new framework for diversification, which extends the state-of-the-art diversification algorithms so as to handle multiple dimensions, and evaluate its performance on a realistic experimental set-up. Our experiments reveal that diversification across multiple dimensions is useful, as shown by improvements in terms of various diversification metrics.

In the second part of the thesis, we exploit supervised learning methods to improve the effectiveness of explicit search result diversification, and introduce three frameworks with different features and goals.

- *LTRDiv*: We formulate the diversification problem as that of learning a ranking model, which is based on the aggregated relevance of each document to all aspects of a given query, and employ two representative LTR algorithms, namely, SVMRank and Random Forests.

- *AspectRanker*: We address a critical sub-task for result diversification and train models (using various query performance predictors as features) to predict the importance of each query aspect. Then, these predicted values are exploited by a traditional (unsupervised) explicit diversification method.

- *LmDiv*: We adapt the LambdaMerge [1] approach for the supervised merging of rankings per query aspect.

We show that our frameworks perform well, and provided impressive improvements over the baselines.

Finally, we address static index pruning from the diversification perspective on the topical context for the first time in the literature. We execute TREC topics over the pruned versus unpruned indexes, and evaluate the retrieved results per query by diversity-aware metrics to expose the effects of pruning. We also propose two novel static index pruning strategies to improve diversification performance by taking into account the topical diversity of documents.

There are several possible future work directions to follow for this thesis. First of all, the multi-dimensional diversification framework can be enhanced by combining diversification and personalization. For instance, given a personalized reading level for a user, it may be still possible to diversify the results in other dimensions (such as document topics and types). We envision that our approach incorporating multiple-dimensions for diversification may also allow such hybrid solutions, where personalization is applied for some dimensions and diversification is applied for the others.

Our supervised learning frameworks can be adapted for different scenarios such as recommender systems. We also plan to investigate the ways of employing alternative diversification metrics as the objective function in the LmDiv framework.

As stated in [99], using query views during static index pruning significantly improves the retrieval effectiveness, hence we can also utilize them in our diversity-aware pruning algorithms. We also plan to apply traditional diversification algorithms (such as xQuAD) to the results obtained over an index pruned with our diversity-aware strategies. Finally, we will explore ways of adopting our diversity-aware methods for the query processing approaches with dynamic pruning.

REFERENCES

[1] D. Sheldon, M. Shokouhi, M. Szummer, and N. Craswell, "Lambdamerge: Merging the results of query reformulations," in *Proceedings of the 4th ACM International Web Search and Data Mining Conference*, pp. 795–804, 2011.

[2] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press, 2008.

[3] B. J. Jansen, A. Spink, and T. Saracevic, "Real life, real users, and real needs: a study and analysis of user queries on the web," *Information Processing and Management*, vol. 36, no. 2, pp. 207–227, 2000.

[4] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong, "Diversifying search results," in *Proceedings of the 2nd ACM International Web Search and Data Mining Conference*, pp. 5–14, 2009.

[5] R. L. Santos, J. Peng, C. Macdonald, and I. Ounis, "Explicit search result diversification through sub-queries," in *Proceedings of the 32nd European Conference on Information Retrieval*, pp. 87–99, Springer, 2010.

[6] R. Santos, C. Macdonald, and I. Ounis, "Search result diversification," *Foundations & Trends in Information Retrieval*, vol. 9, no. 1, 2015.

[7] K. D. Naini, I. S. Altingovde, and W. Siberski, "Scalable and efficient web search result diversification," *ACM Transactions on the Web*, vol. 10, no. 3, 2016.

[8] L. Xia, J. Xu, Y. Lan, J. Guo, W. Zeng, and X. Cheng, "Adapting markov decision process for search result diversification," in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, p. 535–544, 2017.

[9] Z. Jiang, Z. Dou, W. X. Zhao, J. Nie, M. Yue, and J. Wen, "Supervised search

result diversification via subtopic attention," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 10, pp. 1971–1984, 2018.

[10] K. Collins-Thompson, P. Hansen, and C. Hauff, "Search as Learning (Dagstuhl Seminar 17092)," *Dagstuhl Reports*, vol. 7, no. 2, pp. 135–162, 2017.

[11] R. Syed and K. Collins-Thompson, "Retrieval algorithms optimized for human learning," in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 555–564, ACM, 2017.

[12] D. Maxwell, L. Azzopardi, and Y. Moshfeghi, "The impact of result diversification on search behaviour and performance," *Information Retrieval Journal*, pp. 1–25, 2019.

[13] G. McDonald, T. Thonet, I. Ounis, J.-M. Renders, and C. Macdonald, "University of Glasgow Terrier team and Naver Labs Europe at TREC 2019 Fair Ranking Track," in *Proceedings of TREC Conference*, 2019.

[14] M. Zehlike, F. Bonchi, C. Castillo, S. Hajian, M. Megahed, and R. Baeza-Yates, "Fa*ir: A fair top-k ranking algorithm," in *Proceedings of the 26th ACM Conference on Information and Knowledge Management*, pp. 1569–1578, 2017.

[15] R. Santos, C. Macdonald, and I. Ounis, "Exploiting query reformulations for web search result diversification," in *Proceedings of the 19th ACM International World Wide Web Conference*, pp. 881–890, 2010.

[16] E. Aktolga, *Integrating Non-Topical Aspects Into Information Retrieval*. PhD thesis, University of Massachusetts, Amherst, 2014.

[17] Y. Zhu, Y. Lan, J. Guo, X. Cheng, and S. Niu, "Learning for search result diversification," in *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 293–302, 2014.

[18] C. L. A. Clarke, N. Craswell, and I. Soboroff, "Overview of the TREC 2009 Web track," in *TREC*, 2009.

[19] O. Chapelle, D. Metlzer, Y. Zhang, and P. Grinspan, "Expected reciprocal rank for graded relevance," in *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, pp. 621–630, 2009.

[20] C. L. Clarke, M. Kolla, G. V. Cormack, O. Vechtomova, A. Ashkan, S. Büttcher, and I. MacKinnon, "Novelty and diversity in information retrieval evaluation," in *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 659–666, 2008.

[21] A. M. Ozdemiray and I. S. Altingovde, "Query performance prediction for aspect weighting in search result diversification," in *Proceedings of the 23rd ACM Conference on Information and Knowledge Management*, pp. 1871–1874, 2014.

[22] A. M. Ozdemiray and I. S. Altingovde, "Explicit search result diversification using score and rank aggregation methods," *Journal of the Association for Information Science and Technology*, vol. 66, no. 6, pp. 1212–1228, 2015.

[23] C. Eickhoff, J. Gwizdka, C. Hauff, and J. He, "Introduction to the special issue on search as learning," *Information Retrieval Journal*, vol. 20, no. 5, pp. 399–402, 2017.

[24] A. Hoppe, P. Holtz, Y. Kammerer, R. Yu, S. Dietze, and R. Ewerth, "Current challenges for studying search as learning processes," in *the 7th Workshop on Learning and Education with Web Data (LILE2018)*, 2018.

[25] K. Collins-Thompson, P. N. Bennett, R. W. White, S. de la Chica, and D. Sontag, "Personalizing web search results by reading level," in *Proceedings of the 20th ACM Conference on Information and Knowledge Management*, pp. 403–412, 2011.

[26] I. M. Azpiazu, N. Dragovic, M. S. Pera, and J. A. Fails, "Online searching and learning: Yum and other search tools for children and teachers," *Information Retrieval Journal*, vol. 20, no. 5, pp. 524–545, 2017.

[27] K. Raman, P. N. Bennett, and K. Collins-Thompson, "Understanding intrinsic

diversity in web search: Improving whole-session relevance," *ACM Transactions on Information System*, vol. 32, no. 4, pp. 20:1–20:45, 2014.

[28] R. Syed and K. Collins-Thompson, "Optimizing search results for human learning goals," *Information Retrieval Journal*, vol. 20, no. 5, pp. 506–523, 2017.

[29] K. Collins-Thompson, S. Y. Rieh, C. C. Haynes, and R. Syed, "Assessing learning outcomes in web search: A comparison of tasks and query strategies," in *Proceedings of the 1st ACM on Conference on Human Information Interaction and Retrieval*, pp. 163–172, 2016.

[30] C. Eickhoff, J. Teevan, R. White, and S. Dumais, "Lessons from the journey: a query log analysis of within-session learning," in *Proceedings of the 7th ACM International Web Search and Data Mining Conference*, pp. 223–232, 2014.

[31] F. Moraes, S. R. Putra, and C. Hauff, "Contrasting search as a learning activity with instructor-designed learning," in *Proceedings of the 27th ACM Conference on Information and Knowledge Management*, pp. 167–176, 2018.

[32] T. Yilmaz, R. Ozcan, I. S. Altingovde, and Ö. Ulusoy, "Improving educational web search for question-like queries through subject classification," *Information Processing and Management*, vol. 56, no. 1, pp. 228–246, 2019.

[33] A. Usta, I. S. Altingövde, I. B. Vidinli, R. Ozcan, and Ö. Ulusoy, "How k-12 students search for learning?: analysis of an educational search engine log," in *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1151–1154, 2014.

[34] I. B. Vidinli and R. Ozcan, "New query suggestion framework and algorithms: A case study for an educational search engine," *Information Processing and Management*, vol. 52, no. 5, pp. 733–752, 2016.

[35] J. Carbonell and J. Goldstein, "The use of MMR, diversity-based reranking for reordering documents and producing summaries," in *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 335–336, 1998.

[36] C. X. Zhai, W. W. Cohen, and J. Lafferty, "Beyond independent relevance: Methods and evaluation metrics for subtopic retrieval," in *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 10–17, 2003.

[37] C. Zhai and J. D. Lafferty, "A risk minimization framework for information retrieval," *Information Processing and Management*, vol. 42, no. 1, pp. 31–55, 2006.

[38] J. Wang and J. Zhu, "Portfolio theory of information retrieval," in *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 115–122, 2009.

[39] Z. Meng, H. Shen, H. Huang, W. Liu, J. Wang, and A. K. Sangaiah, "Search result diversification on attributed networks via nonnegative matrix factorization," *Information Processing and Management*, vol. 54, no. 6, pp. 1277–1291, 2018.

[40] C. Carpineto, M. D'Amico, and G. Romano, "Evaluating subtopic retrieval methods: Clustering versus diversification of search results," *Information Processing and Management*, vol. 48, no. 2, pp. 358–373, 2012.

[41] H. Yu, A. Jatowt, R. Blanco, H. Joho, J. M. Jose, L. Chen, and F. Yuan, "Revisiting the cluster-based paradigm for implicit search result diversification," *Information Processing and Management*, vol. 54, no. 4, pp. 507–528, 2018.

[42] V. Dang and W. B. Croft, "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*, pp. 65–74, 2012.

[43] S. Hu, Z. Dou, X. Wang, T. Sakai, and J.-R. Wen, "Search result diversification based on hierarchical intents," in *Proceedings of the 24th ACM Conference on Information and Knowledge Management*, pp. 63–72, 2015.

[44] Z. Dou, S. Hu, K. Chen, R. Song, and J.-R. Wen, "Multi-dimensional search result diversification," in *Proceedings of 4th ACM International Web Search and Data Mining Conference*, pp. 475–484, 2011.

[45] S. Hu, Z. Dou, X. Wang, and J. Wen, "Search result diversification based on query facets," *Journal of Computer Science and Technology*, vol. 30, no. 4, pp. 888–901, 2015.

[46] Y. Wang, Z. Luo, and Y. Yu, "Learning for search results diversification in twitter," in *Proceedings of the 17th International Conference on Web-Age Information Management*, pp. 251–264, Springer, 2016.

[47] H. Yu, A. Jatowt, R. Blanco, H. Joho, and J. M. Jose, "An in-depth study on diversity evaluation: The importance of intrinsic diversity," *Information Processing and Management*, vol. 53, no. 4, pp. 799–813, 2017.

[48] Z. Dou, R. Song, X. Yuan, and J. Wen, "Are click-through data adequate for learning web search rankings?," in *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, pp. 73–82, 2008.

[49] S. Vargas, L. Baltrunas, A. Karatzoglou, and P. Castells, "Coverage, redundancy and size-awareness in genre diversity for recommender systems," in *Proceedings of the 8th ACM Conference on Recommender Systems*, pp. 209–216, 2014.

[50] T. D. Noia, J. Rosati, P. Tomeo, and E. D. Sciascio, "Adaptive multi-attribute diversity for recommender systems," *Information Sciences*, vol. 382, pp. 234–253, 2017.

[51] B. Goynuk and I. S. Altingovde, "Supervised learning methods for diversification of image search results," in *Proceedings of the 42nd European Conference on Information Retrieval*, pp. 158–165, Springer, 2020.

[52] D. Carmel, H. Roitman, and N. Zwerdling, "Enhancing cluster labeling using wikipedia," in *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 139–146, 2009.

[53] J. Cohen, "Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit," *Psychological Bulletin*, vol. 70, no. 4, pp. 213–220, 1968.

[54] W. Zheng and H. Fang, "A comparative study of search result diversification methods," in *Proceedings of the 33rd European Conference on Information Retrieval*, pp. 55–62, 2011.

[55] T. Sakai, N. Craswell, R. Song, S. Robertson, Z. Dou, and C.-Y. Lin, "Simple evaluation metrics for diversified search results.," in *EVIA@ NTCIR*, pp. 42–50, Citeseer, 2010.

[56] X. Wang, Z. Dou, T. Sakai, and J.-R. Wen, "Evaluating search result diversity using intent hierarchies," in *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, (New York, NY), pp. 415–424, ACM, 2016.

[57] Z. Dou, R. Song, J. R. Wen, and X. Yuan, "Evaluating the effectiveness of personalized web search," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 8, pp. 1178–1190, 2009.

[58] X. Bai, B. B. Cambazoglu, F. Gullo, A. Mantrach, and F. Silvestri, "Exploiting search history of users for news personalization," *Information Sciences*, vol. 385, no. C, pp. 125–137, 2017.

[59] J. A. Aslam and M. Montague, "Models for metasearch," in *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 276–284, 2001.

[60] D. Carmel and E. Yom-Tov, *Estimating the Query Difficulty for Information Retrieval*. Synthesis Lectures on Information Concepts, Retrieval, and Services, Morgan & Claypool Publishers, 2010.

[61] J. A. Shaw and E. A. Fox, "Combination of multiple searches," in *Proceedings of TREC Conference*, pp. 243–252, 1994.

[62] C.-J. Lee, Q. Ai, W. B. Croft, and D. Sheldon, "An optimization framework for merging multiple result lists," in *Proceedings of the 24th ACM Conference on Information and Knowledge Management*, pp. 303–312, 2015.

[63] Z. Zhang, C. Xu, and S. Wu, "Evaluation of score standardization methods for web search in support of results diversification," in *Proceedings of the Interna-*

tional Conference on Web Information Systems and Applications, pp. 182–190, 2018.

[64] Y. Yue and T. Joachims, "Predicting diverse subsets using structural svms," in *Proceedings of the 25th International Conference on Machine Learning*, pp. 1224–1231, 2008.

[65] S. Liang, Z. Ren, and M. de Rijke, "Personalized search result diversification via structured learning," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 751–760, 2014.

[66] J. Wu, J. X. Huang, and Z. Ye, "Learning to rank diversified results for biomedical information retrieval from multiple features," *Biomedical engineering online*, vol. 13, no. 2, p. S3, 2014.

[67] J. Xu, L. Xia, Y. Lan, J. Guo, and X. Cheng, "Directly optimize diversity evaluation measures: a new approach to search result diversification," *ACM Transactions on Intelligent Systems and Technology*, vol. 8, no. 3, p. 41, 2017.

[68] M. Collins, "Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms," in *Proceedings of the 7th Conference on Empirical Methods in Natural Language Processing*, pp. 1–8, 2002.

[69] Y. Li, H. Zaragoza, R. Herbrich, J. Shawe-Taylor, and J. S. Kandola, "The perceptron algorithm with uneven margins," in *Proceedings of the 19th International Conference on Machine Learning*, pp. 379–386, 2002.

[70] L. Xia, J. Xu, Y. Lan, J. Guo, and X. Cheng, "Modeling document novelty with neural tensor network for search result diversification," in *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 395–404, 2016.

[71] H.-T. Zheng, Z. Wang, and X. Xiao, "A learning approach to hierarchical search result diversification," in *Proceedings of the 1st Asia-Pacific Web and Web-Age Information Management Joint Conference on Web and Big Data*, pp. 303–310, 2017.

[72] S. Cronen-Townsend, Y. Zhou, and W. B. Croft, "Predicting query performance," in *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 299–306, 2002.

[73] A. Shtok, O. Kurland, D. Carmel, F. Raiber, and G. Markovits, "Predicting query performance by query-drift estimation," *ACM Transactions on Information Systems*, vol. 30, no. 2, pp. 11:1–11:35, 2012.

[74] B. He and I. Ounis, "Inferring query performance using pre-retrieval predictors," in *Proceedings of the 11th International Symposium on String Processing and Information Retrieval*, pp. 43–54, Springer, 2004.

[75] C. Hauff, V. Murdock, and R. Baeza-Yates, "Improved query difficulty prediction for the web," in *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, pp. 439–448, 2008.

[76] F. Diaz, "Performance prediction using spatial autocorrelation," in *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 583–590, 2007.

[77] Y. Zhao, F. Scholer, and Y. Tsegay, "Effective pre-retrieval query performance prediction using similarity and variability evidence," in *Proceedings of the 30th European Conference on Information Retrieval*, pp. 52–64, 2008.

[78] Y. Zhou and W. B. Croft, "Query performance prediction in web search environments," in *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, (New York, NY, USA), pp. 543–550, ACM, 2007.

[79] G. Markovits, A. Shtok, O. Kurland, and D. Carmel, "Predicting query performance for fusion-based retrieval," in *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, pp. 813–822, ACM, 2012.

[80] R. L. T. Santos, C. Macdonald, and I. Ounis, "Intent-aware search result diversification," in *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 595–604, 2011.

[81] R. L. T. Santos, C. Macdonald, and I. Ounis, "Selectively diversifying web search results," in *Proceedings of the 19th ACM Conference on Information and Knowledge Management*, pp. 1179–1188, 2010.

[82] T.-Y. Liu, "Learning to rank for information retrieval," *Foundations and Trends® in Information Retrieval*, vol. 3, no. 3, pp. 225–331, 2009.

[83] K. Järvelin and J. Kekäläinen, "Ir evaluation methods for retrieving highly relevant documents," in *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 41–48, 2000.

[84] C. J. C. Burges, R. Ragno, and Q. V. Le, "Learning to rank with nonsmooth cost functions," in *Proceedings of the 20th Conference on Neural Information Processing Systems*, pp. 193–200, 2006.

[85] C. J. Burges, "From ranknet to lambdarank to lambdamart: An overview," *Learning*, vol. 11, no. 23-581, p. 81, 2010.

[86] P. Donmez, K. M. Svore, and C. J. Burges, "On the local optimality of lambdarank," in *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 460–467, 2009.

[87] O. Chapelle, S. Ji, C. Liao, E. Velipasaoglu, L. Lai, and S.-L. Wu, "Intent-based diversification of web search results: metrics and algorithms," *Information Retrieval Journal*, vol. 14, no. 6, pp. 572–592, 2011.

[88] R. L. T. Santos, C. Macdonald, and I. Ounis, "On the role of novelty for search result diversification," *Information Retrieval Journal*, vol. 15, no. 5, pp. 478–502, 2012.

[89] B. T. Dinçer, C. Macdonald, and I. Ounis, "Hypothesis testing for the risk-sensitive evaluation of retrieval systems," in *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 23–32, 2014.

[90] D. Yin, Y. Hu, J. Tang, T. D. Jr., M. Zhou, H. Ouyang, J. Chen, C. Kang, H. Deng, C. Nobata, J. Langlois, and Y. Chang, "Ranking relevance in yahoo

search," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 323–332, 2016.

[91] G. V. Cormack, M. D. Smucker, and C. L. A. Clarke, "Efficient and effective spam filtering and re-ranking for large web datasets," *Information Retrieval Journal*, vol. 14, no. 5, pp. 441–465, 2011.

[92] S. Kharazmi, F. Scholer, D. Vallet, and M. Sanderson, "Examining additivity and weak baselines," *ACM Transactions on Information Systems*, vol. 34, no. 4, pp. 23:1–23:18, 2016.

[93] M. Akcay, I. S. Altingovde, C. Macdonald, and I. Ounis, "On the additivity and weak baselines for search result diversification research," in *Proceedings of the 3rd ACM SIGIR International Conference on Theory of Information Retrieval*, pp. 109–116, 2017.

[94] R. Deveaud, J. Mothe, M. Z. Ullah, and J. Nie, "Learning to adaptively rank document retrieval system configurations," *ACM Transactions on Information Systems*, vol. 37, no. 1, pp. 3:1–3:41, 2019.

[95] J. Lin and C. Dyer, *Data-Intensive Text Processing with MapReduce*. Morgan and Claypool Publishers, 2010.

[96] Y. Tsegay, A. Turpin, and J. Zobel, "Dynamic index pruning for effective caching," in *Proceedings of the 16th ACM Conference on Information and Knowledge Management*, pp. 987–990, ACM, 2007.

[97] D. Carmel, D. Cohen, R. Fagin, E. Farchi, M. Herscovici, Y. S. Maarek, and A. Soffer, "Static index pruning for information retrieval systems," in *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 43–50, 2001.

[98] R. Blanco and A. Barreiro, "Probabilistic static pruning of inverted files," *ACM Transactions on Information Systems*, vol. 28, pp. 1:1–1:33, Jan. 2010.

[99] I. S. Altingovde, R. Ozcan, and O. Ulusoy, "Static index pruning in web search engines: Combining term and document popularities with query views," *ACM Transactions on Information Systems*, vol. 30, no. 1, pp. 2:1–2:28, 2012.

[100] R.-C. Chen and C.-J. Lee, "An information-theoretic account of static index pruning," in *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 163–172, ACM, 2013.

[101] R.-C. Chen, C.-J. Lee, and W. B. Croft, "On divergence measures and static index pruning," in *Proceedings of the 1st ACM SIGIR International Conference on Theory of Information Retrieval*, pp. 151–160, ACM, 2015.

[102] E. S. De Moura, C. F. dos Santos, D. R. Fernandes, A. S. Silva, P. Calado, and M. A. Nascimento, "Improving web search efficiency via a locality based static pruning method," in *Proceedings of the 14th International Conference on World Wide Web*, pp. 235–244, ACM, 2005.

[103] E. S. d. Moura, C. F. d. Santos, B. D. s. d. Araujo, A. S. d. Silva, P. Calado, and M. A. Nascimento, "Locality-based pruning methods for web search," *ACM Transactions on Information Systems*, vol. 26, no. 2, pp. 9:1–9:28, 2008.

[104] R. Blanco and Á. Barreiro, "Static pruning of terms in inverted files," in *Proceedings of the 29th European Conference on Information Retrieval*, pp. 64–75, Springer, 2007.

[105] S. E. Robertson, "The probability ranking principle in ir," *Journal of documentation*, vol. 33, no. 4, pp. 294–304, 1977.

[106] C. J. V. Rijsbergen, *Information Retrieval*. Newton, MA, USA: Butterworth-Heinemann, 2nd ed., 1979.

[107] S. Büttcher and C. L. A. Clarke, "A document-centric approach to static index pruning in text retrieval systems," in *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, pp. 182–189, 2006.

[108] C. Carpineto, R. de Mori, G. Romano, and B. Bigi, "An information-theoretic approach to automatic query expansion," *ACM Transactions on Information Systems*, vol. 19, no. 1, pp. 1–27, 2001.

[109] L. Zheng and I. J. Cox, "Entropy-based static index pruning," in *Proceedings of the 31st European Conference on Information Retrieval*, pp. 713–718, Springer, 2009.

[110] S. K. Vishwakarma, K. I. Lakhtaria, D. Bhatnagar, and A. K. Sharma, "An efficient approach for inverted index pruning based on document relevance," in *Proceedings of the 4th International Conference on Communication Systems and Network Technologies*, pp. 487–490, IEEE, 2014.

[111] L. T. Nguyen, "Static index pruning for information retrieval systems: A postingbased approach," in *SIGIR 2009 Workshop on Large-Scale Distributed Information Retrieval*, pp. 25–32, 2009.

[112] S. Garcia, *Search engine optimisation using past queries*. PhD thesis, RMIT University, 2007.

[113] A. Ntoulas and J. Cho, "Pruning policies for two-tiered inverted index with correctness guarantee," in *Proceedings of the 30th Annual International ACM SIGIRauthor = Wang, Xiaojie and Dou, Zhicheng and Sakai, Tetsuya and Wen, Ji-Rong, Conference on Research and Development in Information Retrieval*, pp. 191–198, 2007.

[114] G. Skobeltsyn, F. Junqueira, V. Plachouras, and R. Baeza-Yates, "Resin: A combination of results caching and index pruning for high-performance web search engines," in *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 131–138, 2008.

[115] R. Baeza-Yates, A. Gionis, F. Junqueira, V. Murdock, V. Plachouras, and F. Silvestri, "The impact of caching on search engines," in *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 183–190, 2007.

[116] Z. Pehlivan, B. Piwowarski, and S. Gançarski, "Diversification based static index pruning-application to temporal collections," *arXiv preprint arXiv:1308.4839*, 2013.

[117] A. Lipani, "On biases in information retrieval models and evaluation," in *ACM SIGIR Forum*, vol. 52, pp. 172–173, ACM New York, NY, USA, 2019.

[118] L. Azzopardi and V. Vinay, "Accessibility in information retrieval," in *European Conference on Information Retrieval*, pp. 482–489, Springer, 2008.

[119] H. Anaya-Sánchez, A. Pons-Porrata, and R. Berlanga-Llavori, "A document clustering algorithm for discovering and describing topics," *Pattern Recognition Letters*, vol. 31, no. 6, pp. 502 – 510, 2010.

[120] H. K. Azad and A. Deepak, "Query expansion techniques for information retrieval: A survey," *Information Processing and Management*, vol. 56, no. 5, p. 1698–1735, 2019.

[121] A. Bouchoucha, J. He, and J.-Y. Nie, "Diversified query expansion using conceptnet," in *Proceedings of the 22nd ACM Conference on Information and Knowledge Management*, pp. 1861–1864, 2013.

[122] E. C. Küçükoğlu, "Search result diversification for selective search," Master's thesis, Middle East Technical University, 2019.

[123] T. Souza, E. Demidova, T. Risse, H. Holzmann, G. Gossen, and J. Szymanski, "Semantic url analytics to support efficient annotation of large scale web archives," in *Proceedings of the 1st International Conference on Semantic Keyword-Based Search on Structured Data Sources*, pp. 153–166, Springer, 2015.

[124] E. Baykan, M. Henzinger, L. Marian, and I. Weber, "A comprehensive study of features and algorithms for url-based topic classification," *ACM Transactions on the Web*, vol. 5, no. 3, 2011.

[125] C. Arya and S. K. Dwivedi, "News web page classification using url content and structure attributes," in *Proceedings of the 2nd International Conference on Next Generation Computing Technologies*, pp. 317–322, IEEE, 2016.

[126] I. S. Altingovde, R. Ozcan, and Ö. Ulusoy, "A practitioner's guide for static index pruning," in *Proceedings of the 31st European Conference on Information Retrieval*, pp. 675–679, Springer, 2009.

[127] R. Blanco and A. Barreiro, "Boosting static pruning of inverted files," in *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 777–778, ACM, 2007.

[128] Z. Dai, C. Xiong, and J. Callan, "Query-biased partitioning for selective search," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pp. 1119–1128, 2016.

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: Yiğit Sert, Sevgi
Nationality: Turkish (TC)
Date and Place of Birth: 1987, Bursa
Marital Status: Married
Phone: +905055914898
E-mail: yigit.sevgi@gmail.com

EDUCATION

Degree	Institution	Year of Grad.
M.S.	Computer Engineering, Ankara University	2012
B.S.	Computer Engineering, Anadolu University	2009
High School	Ahmet Erdem Anadolu Lisesi (MPAL), Bursa	2005

PROFESSIONAL EXPERIENCE

Year	Place	Enrollment
2009-Present	Ankara University	Research Assistant

PUBLICATIONS

From Ph.D. Thesis

Yigit-Sert, S., Altingovde, I. S., Macdonald, C., Ounis, I., Ulusoy, Ö. *Supervised approaches for explicit search result diversification*, Information Processing and Management, 57(6), 102356, 2020.

Yigit-Sert, S., Altingovde, I. S., Macdonald, C., Ounis, I., Ulusoy, Ö. *Explicit diversification of search results across multiple dimensions for educational search*, Journal of the Association for Information Science and Technology.
https://doi.org/10.1002/asi.24403.

Others

Halici E., Bostanci E., Yigit-Sert S. Guzel M. S., Kanwal, N. *The Use of Interpolation Methods for Modelling Human Body Motion*, Human-Computer Interaction, Nova Science Publishers, USA, 2020.

Yiğit-Sert S., Ar Y., Bostancı G. E. *Evolutionary approaches for weight optimization in collaborative filtering-based recommender systems*, Turkish Journal of Electrical Engineering and Computer Sciences, 27(3), 2121-2136, 2019.

Yigit-Sert, S., Kullu, P. *Software cost estimation using enhanced artificial bee colony algorithm*, International Journal of Advanced Computer Science and Applications, 9(4), 2018.

Ar Y., Ünal M., Yigit-Sert S., Bostanci E., Kanwal N., Güzel M. *Evolutionary fuzzy adaptive motion models for user tracking in augmented reality applications*, The 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies, IEEE, Ankara, 2018.

Ozkan O. M., Kullu P., Yigit-Sert S. *Brain Tumor Extraction with Different Segmentation Methods*, The 2nd International Conference on Theoretical and Applied Computer Science and Engineering, Istanbul, 2018.

Bostanci, E., Ar, Y., Yigit-Sert, S. *A Thread Building Blocks based parallel genetic algorithm framework*, International Journal of Research and Reviews in Applied Sciences, 31(1), 1-8, 2017.

Yigit-Sert S., Altingovde I. S., Ulusoy Ö. *Towards detecting media bias by utilizing user comments*, The 8th ACM Conference on Web Science, ACM, Hannover, 2016.

Yigit S., Tugrul B., Celebi F. V. *A complete CAD model for type-I quantum cascade lasers with the use of artificial bee colony algorithm*, Journal of Artificial Intelligence, 5(2), 76-84, 2012.

Çelebi F.V., Yücel M., Yiğit S. *Optical gain modelling in type I and type II quantum cascade lasers by using adaptive neuro-fuzzy inference system*, The 20th Signal Processing and Communications Applications Conference, IEEE, Fethiye, 2012.

Çelebi F. V., Yiğit S. *Tip I ve Tip II Kuantum Kaskat Lazerlerdeki Kırınım İndis Değişiminin Modellenmesine Ait Yeni Bir Yaklaşım*, 7. Ankara Matematik Günleri, Ankara, 2012.

Yiğit S., Eryiğit R., Çelebi F. V. *Optical gain model proposed with the use of artificial neural networks optimised by artificial bee colony algorithm*, Optoelectronics Advanced Materials - Rapid Communication, 5(9), 1026-1029, 2011.

Çelebi F. V., Yiğit S. *Kuantum Kaskat Lazerlerdeki Optik Kazancın Yapay Arı Kolonisi Algoritması Kullanılarak Modellenmesi*, 6. Ankara Matematik Günleri, Ankara, 2011.