Learning to Rank for e-Commerce Search

Fatemeh Sarvi

Learning to Rank for e-Commerce Search

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof. dr. ir. P.P.C.C. Verbeek
ten overstaan van een door het College voor Promoties ingestelde
commissie, in het openbaar te verdedigen in
de Agnietenkapel
op woensdag 22 oktober 2025, te 10:00 uur

door

Fatemeh Sarvi

geboren te Darmeyan

Promotiecommissie

Promotor:	prof. dr. M. de Rijke	Universiteit van Amsterdam
Co-promotor:	prof. dr. S. Schelter	Technische Universität Berlin
Overige leden:	dr. A.J. Biega	Max Planck Institute for Security and Privacy
	dr. D.P. Graus	Universiteit van Amsterdam
	prof. dr. P.T. Groth	Universiteit van Amsterdam
	prof. dr. ir. D. Hiemstra	Radboud Universiteit
	prof. dr. E. Kanoulas	Universiteit van Amsterdam

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

The research was carried out at the Information Retrieval Lab of the University of Amsterdam, funded by Ahold Delhaize.

Cover by VectorMine - shutterstock.com
Printed by Ridderprint, Amsterdam

ISBN: 978-94-6510-901-5

Acknowledgements

Doing a PhD was exciting. But also, not always easy! I started research in a new field, in a new country. At the beginning, it sometimes felt like walking a new path with my eyes closed, but fortunately I was surrounded by many people who helped me tremendously to navigate my way.

Maarten, thank you for investing in me, trusting me, and helping me trust my own ideas. Your guidance has shaped me as a researcher, and I'm grateful for that. Thank you for making sure I found my way. I couldn't have asked for a more caring and supportive supervisor.

Sebastian, your guidance and kindness made every challenge easier. You helped me more than you can imagine, always seeing my work through a different lens.

I would like to thank Asia, David, Djoerd, Evangelos, and Paul for dedicating their time to read my thesis and for kindly serving on my committee.

I was lucky to work alongside so many brilliant and inspiring people at IRLab. I'm grateful to everyone who was part of the lab during my PhD: Amin, Amir, Ana, Andrew, Anna, Antonios, Arian, Ali A, Ali V, Barrie, Chang, Christof, Clara, Clemencia, Chuan, Dan, David, Fei, Gabriel, Gabrielle, Georgios, Hamid, Harrie, Hinda, Hongyu, Ilias, Ilya, Ivana, Jiahuan, Jie, Jingfen, Jingwei, Jin, Justine, Julien, Maarten M, Maartje, Mahsa, Maria, Mariya, Marzieh, Maurits, Ming, Mohammad, Mounia, Mostafa, Mozhdeh, Negin, Nikos, Olivier, Panagiotis, Pablo, Peilei, Pengjie, Philipp, Pooya, Petra, Rolf, Romain, Roxana, Ruben, Ruqing, Sam, Sami, Shaojie, Shashank, Shubha, Songgaojun, Spyretta, Stefan, Svitlana, Thilina, Thong, Vaishali, Vera, Weija, Wanyu, Xinyi, Yang, Yangjun, Yibin, Yifan, Yifei, Yuanna, Yuanxing, Yuyue, Zahra, Zihan and Ziming.

I'd like to especially thank a few colleagues who also became good friends: Ali V and Mozhdeh, thank you for being my paranymphs. Mohammad, thank you for the close collaboration during my PhD. I do miss the last-minute paper rush, though not so much the "one more box plot" moments!

I had the opportunity to be part of AIRLab during my PhD and enjoyed having Barrie, Mariya, Ming, Mozhdeh, Olivier, Sami, Shubha and Stefan as my lab mates. Thank you for all the moments we shared. Through my connection with Ahold Delhaize, I had the chance to meet and work with amazing people. I'd like to thank Almer, Lois and Isaac who helped me at the beginning of my journey at Bol.com; Bart and Bassem who always supported us in collaborations with the companies; and my former and current colleagues Bouke, Anca and Dung, who assisted me during my research internship at Albert Heijn.

I am truly grateful for the incredible friends in my life: Mahsa, Atieh, Ali A, Morteza, Sara, Zahra, Najme, Samira, Navid, Matin and Zeynab. Thank you for multiplying the joy and being there in rough times.

Finally, I would like to thank my family, who were and are my true source of energy and strength. Maman, your endless love and encouragement are the backbone of this

success. Thank you for enduring my absence all these years. Baba, through every moment we shared, you made sure I knew you were proud of me. Thanks to you, I now know that feeling so well and can deeply enjoy and cherish your love. To my siblings, to whom I owe more than they know.

Masoud, I am most grateful to you. You literally adapted your life to make sure I succeeded. Thank you for being incredibly wise, kind, patient, and supportive all these years. You are the constant light when things feel dark, my true partner in life.

Arezoo
Amsterdam, August 2025

Contents

1 Introduction

One of the core challenges in information retrieval (IR) is ranking a collection of data entities, such as documents, multimedia or commercial products, based on their relevance to a specific information need as expressed, for instance, in a textual query [17]. Online search engines and recommendation systems have become essential tools for addressing user information needs. A key element within these systems is the ranker, which orders documents according to their relevance to the user's query [143]. Research in the IR community has increasingly focused on the use of machine learning to develop robust ranking models, known as learning to rank (LTR) methods [92].

In this thesis, we focus on product ranking. This is a specific form of ranking in which user needs extend beyond information discovery. Virtually all major retailers operate their own product search engines, with popular platforms handling millions of search requests daily [77]. E-commerce platforms often provide extensive choices organized within numerous categories, making it nearly impossible for users to find desired items without an effective search engine [137]. The goal of LTR models for product search is to increase user satisfaction by identifying the most relevant products and minimizing the barriers involved in searching, discovering and purchasing items [77, 156].

Product search has unique characteristics that separate it from general web search. One of these characteristics is the way products are displayed on e-commerce platforms. Unlike web search, which relies primarily on text-based documents, product search involves a diverse set of features, including textual (e.g., title and description), numerical (e.g., price and user ratings), and visual elements (e.g., images and star ratings). The presence of textual information, which is only one aspect of product representation in product search, introduces unique challenges. Textual data, such as product titles and queries, are often brief and lack the structure of full sentences, instead consisting of phrases or simple keyword combinations. This limits the effectiveness of traditional lexical matching techniques, such as BM25 [131], which rely on richer text data.

Recent learning-to-match methods have improved web search by capturing semantic similarities between queries and documents rather than relying only on exact term matching [137, 181]. Using techniques such as word embeddings and neural networks, these models can better understand the context, allowing them to link related terms and improve relevance even without exact matches [105, 114]. However, the limited textual data in product search introduces challenges when applying these models, which were

originally designed for web search. This thesis provides insights that help e-commerce practitioners choose high-performance methods suited to the unique demands of product search.

Another unique aspect of product search, particularly on so-called two-sided e-commerce platforms, is the need to satisfy two distinct user groups: customers and providers [138]. Customers, who are end-users looking for relevant and appealing products to purchase, seek relevant items that meet their needs. Providers or suppliers of the products being sold aim to maximize the exposure of their products in search results, as greater visibility can lead to interaction and ultimately increase revenue [156]. Therefore, when an LTR model aims to optimize the rankings to satisfy both customers and providers, accurate exposure estimation becomes crucial. Many factors can influence how items are exposed on search engine result pages. An important factor is the position of the item in the ranked list. The well-known position-based model (PBM) suggests that items in higher positions receive more user attention and thus more exposure [16, 72]. However, position is not the only factor. Inter-item dependencies, i.e., how items relate to each other and stand out from one another, also play a significant role in shaping exposure distribution [16, 138].

In this thesis, we examine a specific case of inter-item dependencies: the effect of an *outlier item* within a ranked list. We define outliers as items that noticeably deviate from others in the list based on certain features, making them stand out and capture user attention [138, 140]. For example, in an e-commerce search, a single item marked with a "Best Seller" tag can be considered as an outlier, as this feature differentiates it from the rest and can draw additional user attention.

We introduce and formalize the concept of outlier items in ranking. Additionally, we explore the impact of outliers on fairness in ranking, assessing how these outlier items affect exposure distribution and, consequently, the fairness towards providers in two-sided e-commerce platforms. Moreover, we investigate whether this shift in exposure distribution alters user click behavior. We propose a method to estimate exposure distribution in the presence of outlier items from the bias in user clicks and correct for this bias to achieve unbiased LTR. Lastly, we analyze the visual saliency of various presentational features and how they influence item outlierness within search results in product search scenario.

1.1 Research Outline and Questions

We give a brief overview of the scope of this thesis and the main research questions that will be answered.

The unique characteristics of product search make it challenging to achieve the same ranking performance as general web search using models specifically designed for web search. As pointed out above, a key challenge is the limited amount of textual data in product search, where product titles and queries are often short and unstructured, typically comprising phrases or simple keyword combinations rather than complete sentences. This limitation amplifies the so-called vocabulary gap compared to other IR tasks [158]. The vocabulary gap arises when documents and queries, represented as bags of words, use different terms to convey the same concepts. While BM25 [131] remains

a robust tool in practical search engines, a growing range of neural learning-to-match methods has been developed to address this gap. These methods go beyond simple lexical matching by embedding queries and documents in finite-dimensional vector spaces and learning their degree of similarity within this space [105, 114]. In Chapter 2 we provide insights into using recent learning-to-match methods for product search by answering the following question:

RQ1 How do learning-to-match models perform in ranking for product search compared to each other in terms of efficiency and accuracy?

We conduct a comprehensive comparison of supervised learning-to-match methods for product search, evaluating both effectiveness and efficiency in terms of training and testing costs. Our experiments involve 12 learning-to-match models and used both public and proprietary datasets, each with $\sim$50,000 queries, to ensure reproducibility and maintain ecological validity [9, 86].

Continuing to explore the distinct characteristics of product search, we focus on factors that influence item exposure in ranked lists, as accurate exposure estimation is crucial for satisfying both user groups, customers and providers, on two-sided platforms. Traditional ranking systems typically order items by relevance to maximize user utility, but in e-commerce, ranking must also ensure fair exposure for providers. Several studies have proposed fair ranking policies to give protected groups, such as various providers, a balanced share of exposure. However, most exposure-based methods [21, 103, 107, 135, 148, 149] assume that exposure is determined solely by position [16, 72], overlooking the impact of inter-item dependencies [16, 138]. In Chapter 3, we hypothesize that a specific type of inter-item dependency, that is, the presence of an outlier product in a ranked list, exists in search logs and can influence exposure distribution. We address the problem of not accounting for this effect by asking the following question:

RQ2 Do outlier items exist in search logs, and how can their effect on exposure-based fair ranking algorithms be mitigated?

We demonstrate the presence of outliers in realistic datasets through an analysis of data from the TREC Fair Ranking track [22]. Additionally, we conduct an eye-tracking study that shows that the order in which users scan items and the exposure each item receives are influenced by the presence of outliers. Based on these findings, we formalize "outliernes" as a new phenomenon within ranking. We propose OMIT, a method designed to mitigate the presence and effects of outliers in ranked lists. With OMIT, it is possible to reduce outliers without compromising user utility or position-based fairness for items.

In Chapter 3, we confirm that outlier items alter exposure distribution in ranked product lists. To effectively use exposure-based models for fair ranking, we propose avoiding outliers and, consequently, their effects on exposure. In Chapter 4 our aim is to estimate exposure distribution rather than avoiding it. We hypothesize that user clicks can serve as a proxy for item exposure and address the following research question:

RQ3 Does outlier bias exist in click data? How can we estimate its impact and correct for this bias?

We conduct a user study to compare the click-through rate (CTR) of specific items under two conditions: displayed as outliers and as non-outliers within the ranked list. Our findings show that CTR is consistently higher when an item is presented as an outlier. Furthermore, an analysis of real-world search logs validates these results, indicating that, on average, outlier items receive significantly more clicks than non-outliers in the same lists. These observations confirm our hypothesis regarding the existence of outlier bias in click data. To correct for this bias, we propose OPBM, a click model based on the examination hypothesis that accounts for both outlier and position bias. We apply regression-based expectation maximization to estimate click propensities using our OPBM model.

In the previous two chapters, we demonstrated that outlier products influence both user attention patterns and click behavior. In those analyses, we either focused on a single product feature or assumed all outlier items had a uniform effect on these shifts, regardless of the specific feature defining their outlier status. However, it is reasonable to expect that users perceive different features differently. For example, a product with a bold "Best Seller" tag might immediately catch more attention than one with a low review score, since users tend to notice the bold discount tag more than a smaller numerical feature [138]. In our final research chapter, we seek to understand more precisely how various presentational features impact users' behavior when examining product lists. We analyze the visual saliency of these features to better understand how attention is distributed across product lists and how specific features influence the stand-out effect of outlier items within a list. This investigation leads us to our last research question:

RQ4 How do different presentational features shape users' perception of outliers and influence exposure distribution in e-commerce search results?

We design visual search experiments to investigate how features like price, star rating, and discount tags affect users' ability to identify outliers, providing initial insights into the immediate observability of these attributes. Next, we conduct eye-tracking experiments to validate and deepen our understanding by observing user behavior in a more realistic, simulated e-commerce environment. We also incorporate visual saliency analysis to predict which product features would naturally attract attention based on their visual properties.

1.2 Main Contributions

This section describes a list of the main contributions in this thesis.

Theoretical contributions

- We introduce, study and formalize the problem of outlierness in ranking and its effects on exposure distribution and fairness (Chapter 3).
- We propose OMIT, an efficient approach that mitigates the outlierness effect on fairness (Chapter 3).

- We identify and study a new type of click bias, originating from inter-item dependencies, called outlier bias (Chapter 4).
- We propose an outlier-aware click model, OPBM, that accounts for outlier items (if they exist), as well as position bias (Chapter 4).

Empirical contributions

- We provide a comprehensive comparison of supervised learning-to-match methods for product search, in terms of both effectiveness and efficiency of training and inference costs, considering 12 learning-to-match method and two datasets (Chapter 2).
- We conduct an extensive eye-tracking user study in two search domains to support our hypothesis about the existence of an effect of outliers on items' exposure (Chapter 3).
- We perform an empirical verification of the effectiveness of OMIT to remove outliers while balancing utility and fairness (Chapter 3).
- We provide an extensive analyses of both user study results and real-world search logs to confirm our hypothesis about the existence of outlier bias in search click data (Chapter 4).
- We empirically show the effectiveness of our outlier-aware model, OPBM, in estimating click propensities, by analysis based on real-world data and semi-synthetic experiments (Chapter 4).
- We demonstrate how different presentational features impact user perception of outlierness in e-commerce search result pages, highlighting the key role of visual complexity in attention distribution (Chapter 5).
- We analyze the influence of bottom-up visual factors on item outlierness in product lists, confirming the effectiveness of the graph-based visual saliency model in detecting visual anomalies in ranked lists (Chapter 5).
- We conduct eye-tracking experiments to demonstrate the impact of top-down factors on user attention, showing that these factors can override bottom-up visual signals in online shopping scenarios (Chapter 5).
- We show that outlier items and their close neighbors in ranked lists attract more attention and receive increased exposure (measured by engagement time), regardless of their position, due to their distinct observable features (Chapter 5).

Resource contributions

- The source code for OMIT (Chapter 3) is released under an open source license; see `https://github.com/arezooSarvi/OMIT_Fair_ranking`.
- The source code for the experiments with OPBM (Chapter 4) is released under an open source license; see `https://github.com/arezooSarvi/outlierbias`.

1.3 Thesis Overview

The thesis starts with the current chapter. In this chapter, we introduce the main subject of the thesis, which is the specifics of applying LTR models to product search. In Chapter 2, we investigate using supervised learning-to-match methods for product search. We conducted a comprehensive comparison of 12 methods for the task of query-product matching and provide insights that help practitioners choose a well-performing method in terms of effectiveness and efficiency in real-world use cases.

In Chapter 3, we discuss fair ranking towards providers in two-sided platforms and show how inter-item dependencies can affect the performance of such algorithms. We introduce the phenomenon of outlierness and study its effect of exposure distribution and subsequently fairness in ranking. Lastly, we propose a novel method to mitigate the negative impact of this effect.

This is followed by Chapter 4, where we estimate the impact of this inter-item dependency, i.e., outlierness, on user examination probabilities. We introduce a new type of click bias, that is, outlier bias. We propose a click model based on the examination hypothesis, which accounts for both outlier and position bias.

Finally, in Chapter 5, we explore how different presentational features influence the perception of outliers in e-commerce search results. We design experiments to measure actual user attention and engagement while examining a product list, providing a more comprehensive view of how outlier features capture and sustain attention in real-world scenarios.

This thesis covers two interconnected but distinct areas of research in LTR for e-commerce search: supervised learning for query-product matching and fairness in ranking. The transition between these areas reflects the natural evolution of my research focus, driven by both theoretical insights and practical challenges in real-world e-commerce platforms.

Chapter 2 focuses on textual data to improve relevance for consumers by optimizing the match between user queries and product descriptions. In this chapter, we address the vocabulary gap, a significant challenge in product search matching. As the research progressed, we realized that the way results are presented and how users perceive them are also crucial factors in product search. Additionally, e-commerce platforms are often two-sided, meaning that rankings impact not only consumers but also providers. This realization led to the second phase of this research, covered in Chapters 3–5. In these chapters, we shift our focus from textual data to observable features and from ranking to user perception. We explore how the interdependencies between the ranked items affect provider fairness, click bias, and user attention. In the conclusion, we offer several suggestions for follow-up research that builds on insights from the two areas of research in LTR for e-commerce search that we pursue in the thesis.

All chapters are based on separate articles. We aim to keep the articles in their original state as much as possible. Because of this, it is unavoidable to have some overlap in the description of some baseline methods or core notation.

1.4 Origins

In this section, we list the publications that form the basis for each chapter:

Chapter 2 is based on the following paper:

- F. Sarvi, N. Voskarides, L. Mooiman, S. Schelter, and M. de Rijke. A comparison of supervised learning to match methods for product search. *SIGIR Workshop on eCommerce*, 2020

  Sarvi: Conceptualization, Investigation, Software, Writing – original draft, Project administration, Formal analysis, Visualization. Voskarides: Investigation, Supervision, Writing – review & editing. Mooiman: Resources. Schelter: Supervision, Writing – review & editing. De Rijke: Funding Acquisition, Supervision, Methodology, Writing – Review & Editing.

Chapter 3 is based on the following paper:

- F. Sarvi, M. Heuss, M. Aliannejadi, S. Schelter, and M. de Rijke. Understanding and mitigating the effect of outliers in fair ranking. In *WSDM*, pages 861–869, 2022.

  Sarvi: Conceptualization, Investigation, Methodology, Software, Writing – original draft, Project administration, Formal analysis, Visualization. Heuss: Methodology, Writing – original draft, Formal analysis, Visualization. Aliannejadi: Supervision, Investigation, Writing – review & editing. Schelter: Supervision, Writing – review & editing. De Rijke: Funding Acquisition, Supervision, Methodology, Writing – Review & Editing.

Chapter 4 is based on the following paper:

- F. Sarvi, A. Vardasbi, M. Aliannejadi, S. Schelter, and M. de Rijke. On the impact of outlier bias on user clicks. In *SIGIR*, pages 18–27, 2023.

  Sarvi: Conceptualization, Investigation, Methodology, Software, Writing – original draft, Project administration, Formal analysis, Visualization. Vardasbi: Formal analysis, Software. Aliannejadi: Supervision, Investigation, Writing – review & editing. Schelter: Supervision, Writing – review & editing. De Rijke: Funding Acquisition, Supervision, Methodology, Writing – Review & Editing.

Chapter 5 is based on the following two papers:

- F. Sarvi, M. Aliannejadi, S. Schelter, and M. de Rijke. How to make an outlier? Studying the effect of presentational features on the outlierness of items in product search results. In *CHIIR*, pages 346–350, 2023.

- F. Sarvi, M. Aliannejadi, S. Schelter, and M. de Rijke. Understanding visual saliency of outlier items in product search. *arXiv preprint arXiv:2503.23596*, 2025

Sarvi: Conceptualization, Investigation, Software, Writing – original draft, Project administration, Formal analysis, Visualization. Aliannejadi: Supervision, Writing – review & editing. Schelter: Writing – review & editing. De Rijke: Funding Acquisition, Supervision, Methodology, Writing – Review & Editing.

The writing of the thesis also benefited from work on the following publications:

- A. Vardasbi, F. Sarvi, and M. de Rijke. Probabilistic permutation graph search: Black-box optimization for fairness in ranking. In *SIGIR*, pages 715–725, 2022.
- M. Heuss, F. Sarvi, and M. de Rijke. Fairness of exposure in light of incomplete exposure estimation. In *SIGIR*, pages 759–769, 2022.
- F. Sarvi. Understanding the effect of outlier items in e-commerce ranking. In *WSDM*, pages 1226–1227, 2023.

2

A Comparison of Supervised Learning to Match Methods for Product Search

The vocabulary gap is a core challenge in information retrieval (IR). In e-commerce applications such as product search, the vocabulary gap is reported to be a bigger challenge than in more traditional application areas in IR, such as news search or web search. As recent learning-to-match methods have made important advances in bridging the vocabulary gap for these traditional IR areas, we investigate their potential in the context of product search. Concerning RQ1, in this chapter we provide insights into the use of learning-to-match methods for product search. We compare both the effectiveness and efficiency of these methods in a product search setting and analyze their performance on two product search datasets, with ∼50,000 queries each. One is an open dataset made available as part of a community benchmark activity at CIKM 2016. The other is a proprietary query log obtained from a European e-commerce platform. This comparison is conducted towards a better understanding of trade-offs in choosing a preferred model for this task. We find that 1. models that have been specifically designed for short text matching, like MV-LSTM and DRMMTKS, are consistently among the top three methods in all experiments; however, taking efficiency and accuracy into account at the same time, ARC-I is the preferred model for real world use cases; and 2. the performance from a state-of-the-art BERT-based model is mediocre, which we attribute to the fact that the text BERT is pre-trained on is very different from the text we have in product search. We also provide insights into factors that can influence model behavior for different types of query, such as the length of retrieved list, and query complexity, and discuss the implications of our findings for e-commerce practitioners, with respect to choosing a well-performing method.

2.1 Introduction

Online shopping is gaining popularity [156]. E-commerce platforms offer rich choices in each of (often) many categories to the point that finding the desired article(s) can be impossible without an adequate search engine. In this context, an effective product

This chapter was published as F. Sarvi, N. Voskarides, L. Mooiman, S. Schelter, and M. de Rijke. A comparison of supervised learning to match methods for product search. *SIGIR Workshop on eCommerce*, 2020.

search engine benefits not just users, but also suppliers.

Implications of the vocabulary gap in product search.. The vocabulary mismatch between query and document poses a critical challenge in search [87]. The vocabulary gap occurs when documents and queries, represented as a bag-of-words, use different terms to describe the same concepts. Although BM25 [131] continues to be a reliable workhorse in practical search engines, there is a growing collection of neural learning-to-match methods aimed specifically at overcoming the vocabulary gap. These methods go beyond lexical matching by representing queries and documents in finite-dimensional vector spaces and learning their degree of similarity in this space [105, 114]. In product search, the vocabulary gap may be a larger problem than in other IR domains [158]. Product titles and queries tend to be short, and titles are not necessarily well-structured sentences, but consist of phrases or simple combinations of keywords.

Semantic matching. While product search leverages a wide range of ranking features [156], features that do not rely on popularity or past interaction behavior are also considered important. Semantic matching is one of the most important techniques to improve the ranking in product search [156, 158]. Several semantic matching methods have already been applied in the area of product search to generate latent representations for queries and product descriptions [156, 189, 190]. Surprisingly, despite recent advances in supervised learning-to-match methods (see Section 2.2 for an overview), relatively little is known about the performance of these methods in the context of product search. In this chapter, we fill this gap by addressing the following research question:

RQ1 How do learning-to-match models perform in ranking for product search compared to each other in terms of efficiency and accuracy?

Our experimental study. We conduct a systematic comparison of 12 supervised learning-to-match methods in the product search task. We compare the ranking performance of these methods in terms of Normalized Discounted Cumulative Gain (NDCG), at positions 5 and 25 (an estimate of first page length in a search session) on two product search datasets, both with more than 50,000 queries. One dataset is an open dataset made available during a benchmarking activity at CIKM 2016, the other dataset is a proprietary dataset obtained from a large European e-commerce platform (Sections 2.3 and 2.4).

Our main experimental finding of this chapter (detailed in Section 2.5) is the following: modern learning-to-match methods are able to make an improvement of 134.46% in terms of NDCG at position 5 of the list, in addition to a lexical baseline that is based on BM25 in the CIKM 2016 dataset, while on the proprietary dataset this improvement is not greater than 29.93%. We attribute this finding to the fact that in our proprietary dataset almost all items presented on the first result page have a high lexical overlap with the query, while in the public dataset the word overlap between the query and the product descriptions is about 1.8%, which is very low [178]. This implies that for the public dataset there are more opportunities for semantic methods to prioritize some items over the others.

We find a high degree of correlation between the performance of the learning to match methods on our two datasets. Except for a special case, ARC-I, we see the same models

corresponding to the top 5 scores achieved for both datasets. We find that models that have been specifically designed for short text matching, such as MV-LSTM and DRMMTKS, are consistently among the top three methods in all experiments, whereas the performance of a BERT-based model is mediocre. Moreover, we show model behavior regarding different aspects of queries, namely query length and popularity, and explain similarities and differences between the two datasets. We also look deeper into the queries for which either of the two matching methods, lexical or semantic, is preferred and discuss their characteristics. We found that for most queries in both datasets semantic matching can improve the ranking, however, since the fraction of queries hurt is substantial, we conclude that query-dependent selection of matching function would be beneficial.

Implications for e-commerce practitioners. The effectiveness in terms of NDCG is not the only criterion in selecting a learning-to-match model for a real-world use case. As Trotman et al. [155] point out, efficiency (both at training and inference time) is a major consideration for product search on e-commerce platforms. We therefore analyze our results with a focus on choosing a suitable model for production deployments, and discuss how this choice is influenced by the trade-off between computation time and model performance (Section 2.6). We have seen that ARC-I provides a good balance between, on the one hand, effectiveness improvements over and above a lexical baseline, with minimum effort required for fine-tuning, and, on the other hand efficiency.

2.2 Related work

Product search. Many approaches have been proposed for product search, ranging from adaptations of general web search models [43] to using versions of faceted search to speed up browsing of products [159, 160]. To help select optimal approaches, Sondhi et al. [150] propose a taxonomy for queries and costumer behavior in product search. Independent of the type of query, it is customary to consider a cascade of two or more steps in producing a search engine Result Page (SERP): first we retrieve all potentially relevant items; then, using one or more re-ranking or learning to rank steps, we decide which items to put on top [155].

There are specific challenges involved in applying learning to rank to product search; Karmaker Santu et al. [77] study these in an e-commerce setting. The signals used for matching queries and products in this learning to rank setup are diverse. E.g., Wu et al. [178] combine three types of feature: 1. statistical features (e.g., total show count, click count, view count, purchase count of a product); 2. query-item features (e.g., content-based query-description matching); and, finally, 3. session features about co-clicked items in sessions. Ludewig and Jannach [95] also consider a broad range of ranking features in a hotel ranking setting, with features ranging from descriptive statistics and latent features to product and location features. In this chapter, we focus, specifically, on query-item features and contrast their effectiveness for product search.

Learning to match. Many learning-to-match methods based on deep neural networks have recently been introduced and used in a range of retrieval tasks; see, e.g., [105, 114] for overviews. In product search, in addition to [95, 178], Bell et al. [18] develop an

e-commerce-specific learning-to-match function based on query specific term weights and Zhang et al. [190] use interaction data between queries and products contained in a graph, along with text embeddings generated by a deep learning-to-match model to rank a list of products. Magnani et al. [98] devise (deep) learning-to-match models for product search, based on different types of text representation and loss functions.

Comparing learning to match methods. Systematic comparisons of (deep) learning-to-match methods are rare. Exceptions include work by Linjordet and Balog [91], who examine the impact of dataset size on training learning-to-match models.

Guo et al. [54] summarize the current status of neural ranking models, as well as their underlying assumptions, major design principles, and learning strategies. They survey the published results of some neural ranking models for ad-hoc retrieval and QA tasks, but mention that it is difficult to compare published results across different papers since even the smallest changes in experimental setup can lead to significant differences.

Brenner et al. [27] contrast the use of learning-to-match models on web search vs. on product search and find a complex trade-off between effectiveness and efficiency. Ludewig et al. [96] benchmark four neural approaches in the context of session-based recommendation against a nearest neighbors-based baseline and identify important lessons for reproducibility. Yang et al. [184] apply five neural ranking models on the Robust04 collection to examine whether neural ranking models improve retrieval effectiveness in limited data scenarios.

What we add on top of the previous work listed above is a systematic study of the effectiveness and efficiency of learning-to-match methods for product search. To facilitate reproducibility, we use the methods implemented in the MatchZoo library [55] for our study, like [91], as well as a BERT-based baseline. We summarize those methods in Section 2.3 of this chapter, and in Sections 2.4 and 2.5 we detail our experimental setup and outcomes.

2.3 Learning To Match Methods

In this chapter, we summarize the learning-to-match methods that we evaluate in our experimental study. Guo et al. [53] propose a categorization of learning-to-match models as follows: *representation-based models* aim to obtain a representation for the text in both queries and documents; *interaction-based models* on the other hand, aim to capture the textual matching pattern between input texts. We follow this organizing principle.

2.3.1 Representation-Based Models

DSSM. The Deep Structured Semantic Model [63] maps text strings to a common semantic space with a deep neural network (DNN) that converts high-dimensional text vectors to a dense representation. Its first layer applies a letter n-gram based word hashing as a linear transformation to reduce the dimensionality of feature vectors and to increase the model's robustness against out-of-vocabulary inputs. Its final layer computes the cosine similarity between the embedding vectors as a measure of their

relevance. This model is trained on clickthrough data to maximize the conditional likelihood of a clicked document given the query.

CDSSM. The Convolutional Deep Structured Semantic Model [146] extends DSSM and adopts multiple convolution layers to obtain semantic representations for queries and documents. Its first two layers transform the input to a representation, based on word and letter n-grams. Next, it extracts local and global (sentence-level) contextual features from a convolution layer followed by max-pooling, before computing the final matching score like DSSM.

MV-LSTM. The MV-LSTM [56] captures local information to determine the importance of keywords at different positions. It leverages a Bi-LSTM to generate positional sentence representations. Its Bi-LSTM generates two vectors that reflect the meaning of the whole sentence from two directions when based on the specified position. The final positional sentence representation is obtained by concatenating these vectors. MV-LSTM produces a matching score by aggregating interaction signals between different positional sentence representations to take into account contextualized local information in a sentence.

ARC-I. The ARC-I [61] network employs a convolutional approach for semantic similarity measurements. It leverages pre-trained word embeddings, and several layers of convolutions and max-pooling to generate separate dense representations for queries and documents. Finally, it applies an MLP to compare the resulting vectors via a non-linear similarity function.

2.3.2 Interaction-Based Models

ARC-II. Models that defer the interaction between inputs until their individual representations "mature," like ARC-I, run the risk of losing information that can be important for matching, because each representation is formed without knowledge of the others [61]. ARC-II [61] addresses this problem by using interactions between query and document, so that the network gets the opportunity to capture various matching patterns between the input texts from the start. It learns directly from interactions rather than from individual representations. A first convolution layer creates combinations of the inputs via sliding windows on both sentences, so that the remaining layers can extract matching features.

DRMM. Guo et al. [53] mention three factors in relevance matching: exact matching signals, query term importance, and diverse matching requirements, and design the architecture of their deep matching model (DRMM) accordingly. DRMM first builds interactions between pairs of words from a query and a document, and subsequently creates a fixed-length matching histogram for each query term. Next, the model employs a feed forward network to produce a matching score, and calculates the final score by a weighted aggregation of the scores of the query terms.

DRMMTKS. This model is a variant of DRMM provided by the MatchZoo [55] library. It is meant for short-text matching, and replaces the matching histogram with a top-k max pooling layer.

MatchPyramid. This convolution-based architecture views text matching as image

recognition [120]. The model first constructs a word-by-word matching matrix by computing pairwise word similarities. This matrix is processed by several convolutional layers to capture the interaction patterns between words, phrases and sentences. In the first layer, a square kernel of size k extracts a feature map from the matching matrix, which is aggregated by max-pooling to fix the feature size. Repetitions of these layers produce higher-level features of a pre-defined size as the final embedding.

K-NRM. The kernel-based neural ranking model employs kernels to produce soft-match signals between words [180]. Given a query and document, it constructs a translation matrix from word pair similarities. These similarities are based on word embeddings that are learned jointly with the ranking model. In the next layer, kernels generate K soft-TF ranking features by counting soft matches between pairs of words from queries and documents at multiple levels. The model combines these soft-TF signals and feeds them into a learning to rank layer to produce the final score.

CONV-KNRM. This model [39] is a variant of KNRM and applies a convolution to represent n-gram embeddings of queries and documents, from which it builds translation matrices between n-grams of different lengths in a unified embedding space. Its remaining architecture is identical to KNRM.

2.3.3 Hybrid Models

DUET. DUET is a duet of two DNNs that combines the strengths of representation- and interaction-based models [106]. DUET calculates the relevance between a query and a document using local and distributed representations and for this reason, has been classified as a hybrid model [54]. By local representation we mean properties like the exact match position and proximity while distributed properties are synonyms, related terms and well-formedness of content.

The local sub-network applies a one-hot encoding to each term, from which it generates a binary matrix indicating each exact match between query and document terms. This interaction matrix is fed into a convolution layer, and the output is passed through two fully-connected layers, a dropout layer and another fully-connected layer, which generates the final output. The distributed sub-network takes a character n-graph based representation [63] of each term in the query and document.

For the distributed part, DUET first learns non-linear transformations to the character-based input by applying a convolution layer on both queries and documents. This step is followed by a max-pooling step, whose output is the processed by a fully-connected layer. The matrix output for the documents is passed through another convolution layer. It then performs the matching by the element-wise product of the two embeddings. Next, it passes the resulting matrix from the previous step to fully-connected layers and a dropout layer until it obtains a single score. These two sub-networks are jointly trained as one deep neural network.

BERT. BERT is a deep bidirectional transformer architecture pre-trained on large quantities of text [41], which has recently achieved state-of-the-art results on ad-hoc retrieval [97, 126]. BERT encodes two meta-tokens, namely [SEP] and [CLS], and uses text segment embeddings to simultaneously encode multiple text segments. [SEP] and [CLS] are used for token separation and making judgments about the text pairs,

respectively. In the original pre-training task [CLS] is used for determining whether the two sentences are sequential, but it can be fine-tuned for other tasks. In our experiments we adopted BERT for classification, and to this end, a linear combination layer is added on top of the classifier [CLS] token [97].

2.4 Experimental Setup

In this section, we introduce our research questions, and describe the two datasets we use for the experiments. We also describe the model tuning procedures and evaluation methodology.

Research questions. Our experimental study aims to answer RQ1 using the following research questions:

RQ1.1 How do learning-to-match models perform in ranking for product search compared to each other and to a lexical matching baseline?

We consider the following aspects while investigating RQ1.1:

- *How do learning-to-match models perform for different types of query in terms of length and popularity?*
- *What is the per query score difference of the best semantic model compared to the lexical matching baseline?*
- *How does BERT perform on short, unstructured text from product search descriptions?*

By answering these questions, we want to provide insight into the usefulness of these models in a comparable setting for product search. In product search, it is important to know the model behavior for different types of query. For example, some e-commerce platforms may prefer to respond as effectively as possible to popular queries even if it yields low performance for long-tailed ones. On the other hand, some might prefer a model that is quite robust to different aspects of queries such as length and popularity. In the last question we want to investigate the impact of BERT pre-trained weights on the data we collected from a product search engine, given the fact that the documents used in pre-training BERT are well-structured sentences, but in our case, the majority of documents (product descriptions and queries) can be considered as phrases or combinations of keywords.

Based on the experimental results, we aim to assist data scientists in e-commerce scenarios by focusing on two additional research questions which concern the choice of a suitable model for e-commerce scenarios:

RQ1.2 Can we come to a general conclusion about which category of models, interaction-based or representation-based, to prefer for the product search task?

Most e-commerce companies have a high number of searches per second; therefore, it is important for them to be able to conduct some part of the model training offline

Table 2.1: Basic statistics of our datasets.

	CIKM 2016	**Proprietary**
#queries	51,888	53,474
#unique queries	26,137	40,125
#unique presented products	37,964	214,778
#clicks	36,814	63,859

for efficiency purposes. Representation-based models can generate embeddings for documents separately from queries (offline), which is an important advantage over interaction-based models in an online setting. This motivates a comparison of these model classes to obtain an understanding of their (dis)advantages for e-commerce practitioners.

Furthermore, every production deployment of machine learning has to take the cost incurred by model training and inference into account, which often has to be traded off against the business benefits provided by the model [90]. The time for inference is also crucial, as the response latency of a model has a major impact on its online performance [12]. We therefore ask the following research question:

RQ1.3 Will the choice of preferred models change when we take training time, required computational resources, and query characteristics into account?

By targeting this question, we aim to assist e-commerce practitioners in making the decision on which models to select as candidates for their production use cases.

Datasets. We conduct experiments on two datasets, one of which is publicly available and the other is proprietary. The basic statistics of the two datasets are shown in Table 2.1.

CIKM 2016. Our first dataset is the publicly released dataset from Track 2 of the CIKM Cup 2016.[1] This dataset contains six months of anonymized user search logs, including query and product description tokens, clicks, views, and purchase records on an e-commerce search engine from January 1st, 2016 to June 1st, 2016. The dataset contains additional product metadata such as product categories, description, and price. In the original split of the data, the last query of each session is marked as a test sample, so we do not have user interaction signals for these queries. In addition to search sessions that come with queries, this dataset also contains browsing logs that are query-less. We ignore this part of the data in our experiments, since we are interested in query/document matching based on text, and we study its impact in a SERP re-ranking task. As a consequence, since the results achieved here are only based on text matching in query-full queries, they are not comparable to studies in which the whole dataset is used to improve the ranking, such as [178, 190].

Proprietary dataset. Our second dataset is extracted from the search logs of a large, popular European e-commerce platform. The main language of the dataset is Dutch. This dataset contains sampled queries from ten days of users' search logs. We leverage the first nine days as training data and the last day queries as test data. For each query,

[1] `https://competitions.codalab.org/competitions/11161`

we know the items rendered on the first result page. We only use the title for each item, which contains a short description of the product to be consistent with the CIKM 2016 data. We label positive matches between queries and items according to observed click-through data, and remove sessions without clicks from the dataset [70]. In the preprocessing step, we remove punctuation marks, HTML tags, and other unknown characters from the text. Also, we lowercase all the tokens.

Model tuning. We experiment with models from the well established MatchZoo [55] library,[2] which itself is based on Keras and TensorFlow. The MatchZoo library contains a tuning module that fine-tunes the models based on pre-defined model-specific hyper-parameter spaces [55]. In the tuning process parameters are sampled from the hyper-space, but this sampling is not random, the scores of past samples will have an effect on the future selection process in a way that it yields a better score. The tuner uses Tree of Parzen Estimators (TPE) [19] to search the hyper-space. We start tuning from the default parameter values provided by MatchZoo, and select hyper-parameters for all models based on a fixed validation set in 50 rounds.

For the experiments with BERT, we used the implementation from Contextualized Embeddings for Document Ranking (CEDR)[3] provided by MacAvaney et al. [97]. We employed BERT base model (12 layers) with multilingual weights as well as a (Dutch) monolingual version called BERTje [40], which is trained on a large and diverse dataset of 2.4 billion tokens. We conducted separate experiments with both sets of weights in this chapter. We also added gradient clipping and warm-up steps from the HuggingFace transformer [173] implementation to improve the performance.[4]

Evaluation setup. The cascade model [37] assumes that users scan items presented in a SERP one by one, from the top of the list, and the scan will continue after observing non-relevant items but stops after a relevant item is found. Motivated by this assumption, we only consider the items above the last clicked product in the list as well as two items below that. This approach additionally helps to reduce the data size while balancing the number of positive and negative samples in our data. This principle is applied to both datasets, and we use the same maximum number of epochs with early stopping for training all the models. Moreover, since we labeled our proprietary data only based on clickthrough information, we treat clicks and purchases identically for the CIKM dataset. While we consider the items to be presented in a list, it is common for e-commerce websites to use a grid view to display the products. In this case, users' examination behavior can be different from the cascade model we use in this study.

Evaluation metrics. We report NDCG at two cut-offs: 5 and 25. We decided for a cut-off of 5 because the top items returned for a query are important to capture a user's attention; we choose 25 which is the maximum number of results per query in both datasets, and it is a good estimate of the items shown on the first page of an e-commerce website.

[2] https://github.com/NTMC-Community/MatchZoo
[3] https://github.com/Georgetown-IR-Lab/cedr
[4] https://github.com/huggingface/transformers

2.5 Performance of Learning to Match Methods for Product Search

In this section, we seek to answer the research questions mentioned in Section 2.4. For this purpose, we study the performance of different learning-to-match models in a comparable setting.

We first address RQ1.1: *How do learning-to-match models perform in ranking for product search compared to each other and to a lexical matching baseline?* by determining the best-performing method or group of methods in bridging the vocabulary gap for product search.

The overall performance of the learning-to-match methods on our datasets is summarized in Table 2.2. Here, we re-rank an original ranked list obtained by a lexical matching method as first step in a two-step retrieval cascade. For the CIKM data, the original ranking comes from BM25, so it is only based on matching between query, and product title. For the proprietary data, we omit signals involved in the production ranking which are not related to lexical match, so that we obtain a similar baseline as the one we have for the public data. As a result, the ranking produced by the lexical baseline is purely based on matches of the query and the title, which enables us to compare the results to the BM25 baseline provided for the public dataset.

Table 2.2: Performance of MatchZoo models on both datasets in terms of NDCG at position 5 and 25.

Model	CIKM data		Proprietary data	
	NDCG@5	@25	@5	@25
Lexical	0.148	0.343	0.314	0.474
MatchPyramid	0.152	0.347	0.287	0.454
CDSSM	0.314	0.452	N/A	N/A
ARC-II	0.320	0.458	0.334	0.488
ARC-I	0.326	0.462	0.408	0.549
DRMM	0.331	0.464	0.288	0.455
DSSM	0.334	0.467	N/A	N/A
KNRM	0.341	0.472	0.337	0.490
DUET	0.345	0.473	0.350	0.500
MV-LSTM	0.342	0.474	0.408	0.549
CONV-KNRM	0.347	0.476	0.349	0.498
DRMMTKS	0.347	0.477	0.345	0.498
Best-BERT	N/A	N/A	0.340	0.493

Results for the CIKM2016 dataset. For the CIKM dataset, all learning-to-match models outperform the lexical match baseline. The score achieved by MatchPyramid is almost the same as the baseline, but other models perform 112.16% to 134.46% better than BM25 in terms of NDCG@5 for the public dataset. It is worth mentioning that, although, the differences in scores might not be noticeable, they indicate improvement

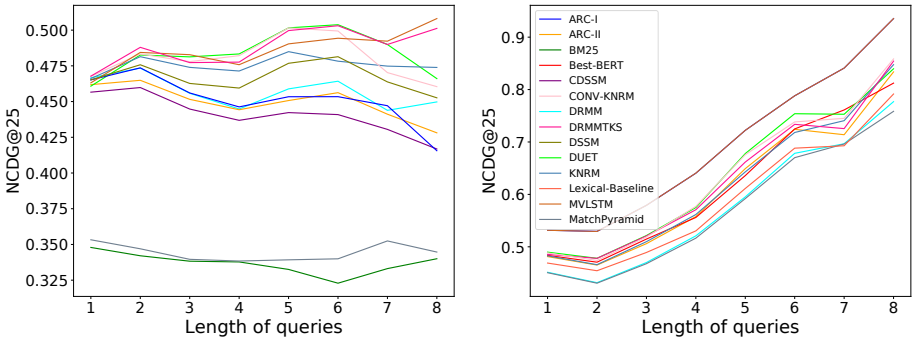
for many queries. In other words, the 0.001 difference between the scores achieved by CONV-KNRM and DRMMTKS at position 25, means better NDCG for 3.22% of test queries. DRMMTKS performs better than the baseline for 36.02% of test queries. This confirms that semantic matching can indeed improve the matching of query/item pairs in product search as well as more general ranking tasks, even in the absence of well-structured sentences or long documents.

Results for the proprietary dataset. Not all semantic matching models outperform the lexical matching baseline for the proprietary dataset. Specifically, MatchPyramid and DRRM achieve lower results for this dataset. On average, the spread of the results we get for this data is smaller than for the CIKM dataset. In the latter the improvement made by the best performing model – DRMMTKS – is roughly 134.46% better in terms of NDCG@5 compared to the lexical baseline, which implies that in this case, semantic matching can greatly help ranking most relevant items on top of the list. However, the corresponding improvement of our learning-to-match methods for the proprietary dataset is not bigger than 29.93% when we take ARC-I/MV-LSTM as the best performing semantic methods. If we compare MatchPyramid's score as the lowest, to ARC-I/MV-LSTM the gain is 42.16% which is still way smaller than what we see in the public data. It should be noted that, although we report the same score for ARC-I and MV-LSTM, they perform differently for 0.4% of test samples. Since the difference is marginal, it is not visible in 3 decimal digits.

We attribute the lower impact of the semantic matching methods on the proprietary dataset to the fact that almost all the items presented on the first page have a high lexical overlap with the query. In other words, the diversity of the first page, if we only consider contextual aspects of products (titles) as the source of diversity, is much smaller compared to the public dataset. This implies that for the public dataset there are more opportunities for semantic methods to prioritize some items over others. Besides, the chronological split of our proprietary dataset makes it a more challenging case than the public dataset.

In general, we observe that models like MV-LSTM and DRMMTKS are consistently among the top performing methods in all experiments, which we attribute to the fact that these models have been specifically designed for short text matching. The average length of queries in our public dataset is 3.1 and the average length for product descriptions is 4.8, which both are very short. Note that we could not successfully finish a run of CDSSM and DSSM on the proprietary dataset, due to out-of-memory issues with the respective MatchZoo implementations. We plan to address this issue in future work.

An aspect of RQ1.1 is to investigate the performance of BERT-based models on short, unstructured text from product search logs. As indicated in Table 2.2, our best performing BERT-based model (Best-BERT) which employed "bert-base-multilingual-uncased" pre-trained weights, and early stopped after 10 patience steps, is not among the top-ranked models for our proprietary dataset. In Section 2.4 we mentioned that we also considered another version of BERT named Bertje, however, since we have English terms mixed in with the (predominantly) non-English text in product titles and queries, multilingual weights performed better than a monolingual model. The NDCG@25 achieved with Bertje pre-trained weights on our dataset is 0.488 while the score achieved from multilingual weights, in the same setting, is 0.493. It is worth



(a) CIKM dataset **(b)** Proprietary dataset

Figure 2.1: Ranking performance for varying query length. On the X-axis we see the length of the query, and Y-axis indicated the average NDCG at position 25 per queries of a specified length.

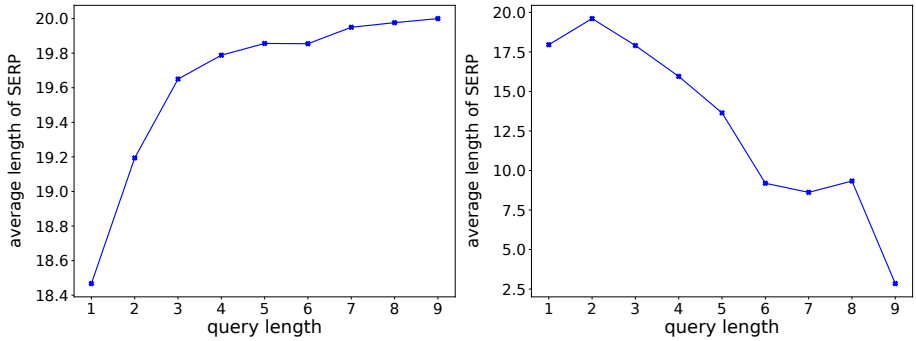
mentioning that to study the impact of fine-tuning on our dataset, we once applied the model on the test data without any fine-tuning. The performance we obtained is 0.445 which implies the effectiveness of fine-tuning. We conjecture that one of the main reasons behind this poor performance from state-of-the-art BERT is the fact that the text it is pre-trained on is very different from the text we have in product search; more investigations are needed to support this conjecture.

2.5.1 Performance for Different Types of Queries

Next, we drill down into the experimental results to investigate an additional aspect of RQ1.1: *How do learning-to-match models perform for different types of query in terms of length and popularity?*

Query length. Figure 2.1 depicts the ranking performance of all models under varying query lengths. Most of the queries in our datasets contain only a single word, but there are a few very long queries with more than 75 words in the CIKM data and smaller queries of 17 words in our proprietary data. Note that we restrict ourselves to query lengths up to 8 words, for which we have a sufficient number of samples in our datasets.

For the proprietary dataset (Figure 2.1b), we observe that as the query length increases, the matching performance also increases. All models follow the same trend. Figure 2.2 shows the average number of items presented in response to queries in both datasets. In general, this number is larger for the CIKM data, and we can see that the length of SERP increases by the length of queries. In our proprietary dataset, however, longer queries result in fewer items, which is attributed to the fact that in most cases these long queries are the exact descriptions of specific products which are already known by the users. Since in these cases, the search engine can precisely retrieve the intended products, users can be easily satisfied, which is visible in high scores of NDCG for these queries. Unfortunately, we do not have access to the actual content of the CIKM queries, so we cannot further interpret the behavior of this data.



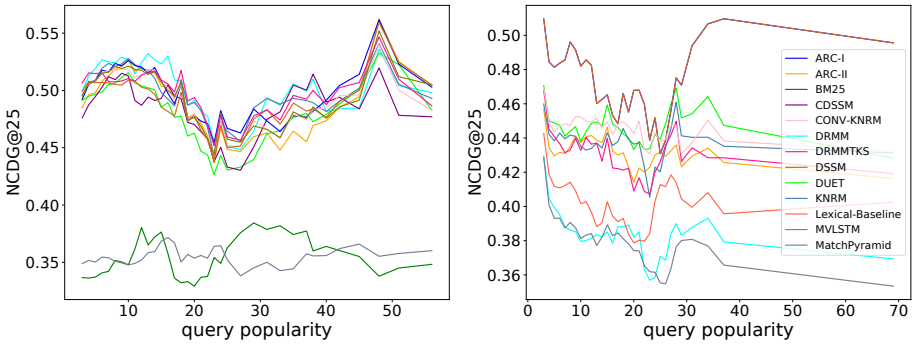
(a) CIKM dataset **(b)** Proprietary dataset

Figure 2.2: Average number of items presented on SERP for different lengths of queries.

For the CIKM dataset (Figure 2.1a) however, we cannot arrive at a reasonable conclusion about the relationship between query length and performance. Since the terms in this dataset are hashed, it is difficult for us to investigate the performance of different methods based on the length of the query because we do not know the original terms. When comparing different models, KNRM seems more robust to query length compared to the other models, and CONV-KNRM also performs quite consistently for shorter queries.

Query popularity. Figure 2.3 depicts the results based on the popularity of queries, i.e., the number of times a query is repeated throughout our dataset. We again only include popularity values for which we have a sufficient number of samples. The X-axis indicates the popularity of the queries, and the Y-axis denotes the average NDCG at position 25.

Interestingly, we observe a "valley" in the middle for both datasets. We can explain what we see in Figure 2.3b in three steps: starting from leftmost part of the plot, it contains less popular queries which are usually longer than the popular ones, and from what was indicated in Figure 2.1b, we know that it is easier for the models to rank items for these types of queries. That is why we see a relatively high performance at this part. However, as the popularity increases the queries get shorter. The middle part contains queries that are repeated between 10 to 15 times, we encounter some shorter queries, which are more challenging for the models, and are not repeated often enough for the models to pick up the patterns between the pairs of these queries with the large number of associated items. This situation gets more difficult since based on the nature of the original production ranking function which was employed during logging of our proprietary data, we often see that the retrieved list for a query can vary a lot from day to day. As we move further in Figure 2.3b toward the most popular queries which we consider to be short, the performance improves. The reason for this observation can be that popular queries get repeated over and over again, the lists of items presented for them converge, and the models can learn the relationship between the pairs. As expected, we see that the models generally perform better for more popular queries.



(a) CIKM dataset **(b)** Proprietary dataset

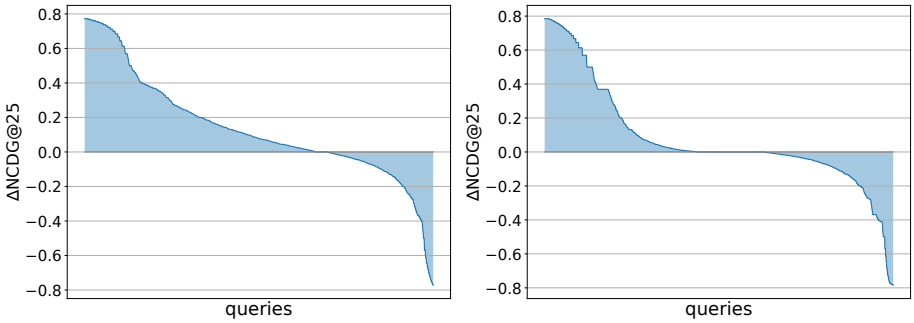
Figure 2.3: The behavior of models based on query popularity. The flow is quite the same in all cases: all models tend to perform better for more frequently seen queries.

2.5.2 Per Query Score Difference between the Best Semantic Model and the Lexical Matching Baseline

Next, we focus on the final aspect of RQ1.1: *What is the per query score difference of the best semantic model compared to the lexical matching baseline?*

Figure 2.4 shows the difference between the best performing semantic model and the lexical match per query for each of the datasets. The best performing semantic model for the CIKM dataset was DRMMTKS, while ARC-I and MV-LSTM performed best for the proprietary data. The Y-axis shows the difference in NDCG at position 25 of the best performing model to the lexical baseline for all test queries. The X-axis lists the queries in decreasing order of ΔNDCG such that the queries for which the semantic model performs better are on the left and vice versa for the lexical model on the right. Queries that benefit from semantic matching have a positive value on the Y-axis while those that prefer lexical matching have a negative value. The plots indicate that semantic matching improves the ranking for most of the queries. This is more obvious in Figure 2.4a, considering that semantic matching has more influence on the CIKM data than for the proprietary data, which is also presented in Table 2.2. Although in Figure 2.4b this difference is not as visible, the area under the curve for the upper part is $\sim$1.2 times bigger than the part below the X-axis. This suggests that query-dependent selection of matching function would be beneficial.

In the case of the CIKM data, it is hard to interpret the queries from the uppermost and bottom points of the plot, since we do not have access to the actual content of the queries. However, for the proprietary data, we can analyze these respective queries. We observed that, analogous to the public dataset, there is no meaningful difference in terms of the length and the popularity of these queries, but the words in the queries for which semantic matching performs best are more general and commonly used than the words, which we see in the other group. For example, among the queries for which semantic method vastly outperforms lexical matching, we encounter queries like "*wireless earbuds*", and "*lego*", which are closer to a category name than to one



(a) CIKM dataset **(b)** Proprietary dataset

Figure 2.4: Per-query paired differences between the best semantic model and lexical baseline for models trained on each dataset and evaluated on the test sets. The Y-axis indicates ΔNDCG at position 25 of ranking between best semantic model and lexical baseline The X-axis lists the queries in the referenced dataset in decreasing order of ΔNDCG such that queries for which semantic model performs better are on the left and vice versa for the lexical model on the right.

specific product. On the other hand, we have examples like "*anneke kaai*," "*buzzed*" and "*Stephan Vanfleteren*" for queries with a higher NDCG achieved by the lexical matching baseline, these are mostly proper nouns or specific items.

There are some queries that are repeated in both groups. In other words, we can see sessions with the same query, while in one session lexical matching performs better, in the other one semantic matching provides a better ranking. This is because of different relevance judgments from different users based on personal preferences. Examples of these queries are "*star wars lego*" or "*perfume*" of different brands. In these cases, all the products rendered in SERP contain the exact words from query, so the only factor that makes a user click on an items is personal preferences and possibly the position of the product in the list. When these preferences vary from user to user there is no discriminating signal that the semantic models can capture to prioritize one item over the other for future queries.

When looking at per query best/worst performances of the semantic matching model and the lexical baseline on our proprietary dataset, we see that for 62.3% of the queries both models either perform accurately or poorly. However, in 15.9% of the cases, we have queries for which the performance of semantic matching is high, while lexical matching does not perform accurately. On the other hand, we see that the opposite behavior, i.e., where the lexical matching baseline outperforms semantic matching, is much rarer (7.9%).

Again it is interesting to look at some examples: among the queries for which semantic method is preferred, we can see queries expressed in general terms like "*woonkamer klok*," "*tractor hout*" or "*panty met print*," which means "living room clock," "tractor wood," and "panty with print," respectively. These queries do not match to any specific item, as they are more exploratory queries. Among the queries that do

not benefit from a semantic method and prefer a lexical match, we again mostly see queries with proper nouns, and more interestingly combinations of numbers like *11 400 700* which matches the sizes of some charging cables. For these queries, the user exactly knows what he/she is looking for in the product catalog and we see that these terms match to parts of titles.

2.5.3 Further Considerations

Impact of word embeddings. For some of the models, the word embeddings used for initialization are more critical than for others. For example, DRMM creates a similarity matrix based on the word embeddings at the beginning, and does not update the embeddings in the training process (like MV-LSTM does, for example). In our experiments, we have all models start with a random initialization in order to have an identical setting for both datasets to make the results comparable. However, as a result we encountered very poor performance for DRMM as one of the worst performing models with NDCG scores of 0.331 and 0.288 at position 5 for the public and proprietary data, respectively, which contrasts other studies, where it proved to be one of the strongest models for different tasks [54, 184].

We ran an additional experiment for DRMM, ARC-I and ARC-II in order to investigate the impact of the leveraged word embeddings on their performance. For training these models we experimented with both a Word2Vec model learned from a large corpus of general text in the same language as our proprietary data, and a Word2Vec model learned from the text of a corresponding proprietary product catalog and our training queries. However, we did not observe meaningful improvements over using pre-trained embeddings.

The problem with using embeddings learned from general text is that queries and product descriptions of our proprietary data contain both English and non-English text, so when using only non-English embeddings the model misses plenty of words. To solve this problem we also trained a Word2Vec model on the product catalog and training queries. However, we found it difficult to balance the amount of product descriptions and queries to achieve robust weights from the Word2Vec model.

2.6 Implications for E-Commerce

Experiments into position bias in e-commerce settings have shown that customers are prejudiced towards the first few results [60]. It is customary to rank products primarily based on the popularity of a product without taking semantics into account [155]. However, user studies and analyses of interaction logs reveal that customers use various queries with subtle differences to search for the same product set that should lead to different rankings [150]. Adding semantics, to query understanding and to product ranking, should result in a better ranking. From the point of view of generating semantically more meaningful ranked lists of products than purely ranking by popularity, our experiments in Section 2.5 suggest that we should consider models like ARC-I, MV-LSTM and DRMMTKS.

But effectiveness as measured in terms of NDCG is not the only criterion in selecting

a learning-to-match model for a real world use case. As Trotman et al. [155] point out, for product search on e-commerce platforms, efficiency is a major consideration: both efficiency at training time and at inference time. In order to accommodate for the special considerations in production use cases, we analyze our results in this section. This discussion can be leveraged as a starting point for deciding which model to choose for a real world deployment by e-commerce practitioners. We focus on the question of which model family to choose (Section 2.6.1) and how this choice is influenced by the trade-off between computation time and model performance (Section 2.6.2).

2.6.1 Choosing between Representation-Based Models and Interaction-Based Models

We now discuss RQ1.2: *Can we come to a general conclusion about which category of models, interaction-based or representation-based, to prefer for the product search task?*

The motivation for this question is as follows. Most e-commerce companies have a high number of searches per second and are at the same time continuously expanding their total number of products, partners, and customers. This results in a lot of new product and interaction data per day, as well as an ever-shifting catalog of products. As a consequence, we require a model that can be extended with new data and is able to rank products that have not been seen with a particular query before.

Representation-based models can generate embeddings for documents separately from queries and we can cache these embeddings for efficiency purposes. In these models the embeddings for query and product are not dependent on each other unlike interaction-based models where query and product are linked. This allows us to easily compute the embeddings for all products and popular queries offline. For interaction-based models, even if we manage to cache the representations of top items retrieved for popular queries, in the case of new products or changed items (possible changes in product description or other contents that we might use), we again need to compute online which can be very time consuming.

Thus, we have a preference for representation based models. Based on the results in Table 2.2 interaction-based models, namely DRMMTKS and CONV-KNRM are the two best performing models for the public dataset with NDCG@25 of 0.477 and 0.476 respectively. However, the representation-based model MV-LSTM is in the third rank with the score of 0.474, which is a marginal decrease compared to the two interactive models. Furthermore, for our proprietary dataset the best performance is from representation-based models, ARC-I and MV-LSTM, which is fortunate for practitioners. In summary, we would recommend representation-based models for production deployments in general, due to the discussed operational advantage of being able to incorporate new query and production representations easily [54].

2.6.2 Training Cost and Inference Speed

Next, we discuss efficiency specific aspects of RQ1.3: *Will the choice of preferred models change when we take training time, required computational resources and query characteristics into account?* In the real world, resources are limited; by answering this

question we aim to assist e-commerce practitioners with the decision which models to select as candidates for their production use cases. Inference time is also important since the response time of the system has a huge impact on the user experience.

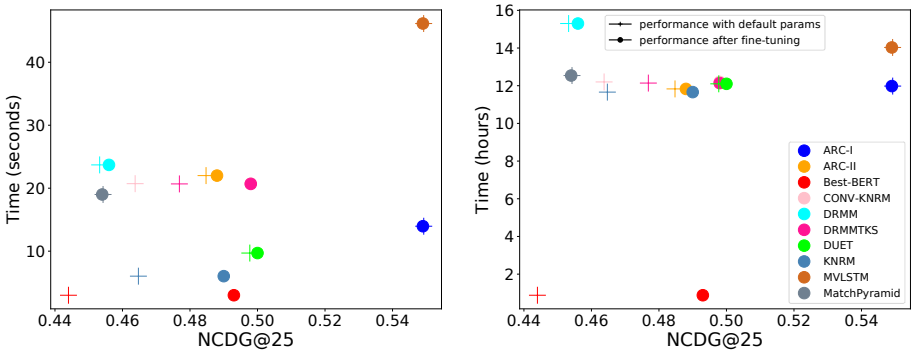
We compare the training and inference time for the learning-to-match models on our proprietary dataset. We only provide this information for our proprietary data, since it contains raw search logs extracted from an e-commerce platform, and the chronological split of test/train samples is a more acceptable representation of the real world use cases. Given that we have to trade-off computation cost and ranking performance, based on Figure 2.5, we conclude that ARC-I and Best-Bert are the strongest candidates, since they have strong ranking performance (see Table 2.2) while having low training and inference times. Considering ARC-I, it is really important that we could achieve the best performance on our dataset using this model with the default values for its hyper-parameters (Figure 2.5). This means that compared to the other models, it is possible to obtain a sufficient performance using ARCI-I without the need to fine-tune the model.

Memory consumption. In Section 2.5 we mentioned that we could not successfully finish a run of CDSSM and DSSM on the proprietary dataset, due to out-of-memory issues with the respective MatchZoo implementations. MatchZoo has specific preprocessing modules for these two models that include word hashing. This preprocessing step consumes a huge amount of memory, which makes it inapplicable of being applied on our proprietary data. Although MatchZoo provides us with the option of not using word hashing for the preprocessing step of the training process, for the evaluation part it does not support the same setting. From the experiments we observed that CDSSM was considerably slower than DSSM, but none of them can be considered proper choice for a big dataset like ours.

It should also be noted that, although the MatchZoo implementation supports GPU computation, we observed a very low GPU usage for all of our models during training and evaluation time. In terms of average computational time spent on GPU non of the models exceeds 5% GPU utilization during the whole process on any of our two datasets. Since we checked the functionality of the implementation with a QA dataset consisting of long documents, we can attribute this observation to the fact that our texts are too short to engage the GPU properly. We also contacted the MatchZoo team and they suggested to increase the batch size to a very big number to solve the issue, but since it could cause a lower ranking performance we did not follow this advice.

2.6.3 Impact of Query Characteristics

Finally, we discuss aspects of RQ1.3 with respect to query characteristics. Query characteristics are important for an e-commerce platform because it is important to know which model is preferred in which case. Length of the query and query popularity are two characteristics that differ per language and per device on the platform. The results from Section 2.5.1 show that query length does have an influence on the model and it could be worth investing into various models for various settings if the average query length differs per setting. For example, from the analysis conducted, we see that customers who access the e-commerce platform through an app use shorter queries than customers who use a browser. E-commerce platforms could develop different models



(a) Performance vs. inference time **(b)** Performance vs. training time

Figure 2.5: Ranking performance in comparison to training and inference time for the proprietary dataset. Both ranking performances achieved from the default hyper-parameters and the fine-tuned ones are depicted in this figure.

for different settings, although this might affect consistency to an unacceptable level.

Query popularity is the second important characteristic. Ideally, each query should result in the best ranking for the customer. However, popular queries are searched more often and thus these influence the revenue more than less popular queries. A model should thus work well enough on the less popular queries and excellent on the popular queries with a certain trade-off, so MV-LSTM seems to be a good choice. Having said that, there exist cases where less popular queries can equally influence the revenue (e.g., when they are issued after a popular query in a session). This should also be taken into consideration when selecting rankers for deployment.

2.6.4 Recommendations for Deployment

In terms of overall performance, we have seen that ARC-I provides a good balance between, on the one hand effectiveness improvements over and above a lexical baseline and on the other hand efficiency. Plus, the fact that it can perform well with the default configuration is another positive point.

Next, we found that, most methods perform consistently across different query lengths on the CIKM dataset, with the results for MV-LSTM going up as query length increases; on the proprietary dataset, the performance of all methods consistently increases with query length. In terms of query popularity, we see consistent performance across different levels of popularity for all learning-to-match models on the CIKM dataset, but on the proprietary data we seen that some learning-to-match methods clearly benefit from increased popularity, including DUET and MV-LSTM.

Furthermore, in a side-by-side comparison of top performing learning-to-match methods and a lexical method, we see that substantial fractions of queries are helped by the learning-to-match method than are hurt, on both the CIKM and the proprietary datasets, while the fraction of queries hurt is substantial. The latter suggests that query-dependent selection of a matching function would be beneficial. Finally, we found that

representation-based models provided the best accuracy and efficiency trade-off.

2.7 Conclusion & Future Work

In this chapter, we have established a comprehensive comparison of supervised learning-to-match methods for product search. We have considered 12 learning-to-match methods, considering both effectiveness and efficiency in terms of training and testing costs. Our comparison was organized around three main research questions using two datasets.

From the experiments, we find that models that have been specifically designed for short text matching, like MV-LSTM and DRMMTKS, are among the best performing models for both datasets. By taking efficiency and accuracy into account at the same time, ARC-I is the preferred model at least for our proprietary data, which is a good representation of the real-world e-commerce scenario. Moreover, the performance from a state-of-the-art BERT-based model is mediocre, which we attribute to the fact that the text BERT is pre-trained on is very different from the text we have in product search. We also provide insights that help practitioners choose a well-performing method for semantic matching in product search.

In the next chapter, we will shift our attention from query-product matching using textual data, to another aspect of e-commerce search, i.e., inter-item dependencies in a ranked list. we will introduce the concept of outlierness in rankings and investigate its effects on exposure distribution and provider fairness. Accordingly, we will raise and address RQ2 in the following chapter.

2.8 Reflections

This study was conducted in 2019 and published in 2020. Since then, the field of IR and consequently, semantic matching has evolved significantly. At the time of this research, supervised deep learning-to-match methods, including BERT [41], were the dominant approach to improve query-product matching. Our work provided insights into how best to leverage these existing methods for the specific case of product search. However, recent advancements in large language models (LLMs) and representation learning for information retrieval have introduced more powerful models that can better capture semantic relationships between a query-document pair.

More advanced large-scale pre-trained Transformer-based models for ranking tasks, such as ColBERT [78] and T5-based models [111, 124, 192] have demonstrated superior performance by leveraging contextualized embeddings and dense retrieval methods, outperforming traditional supervised learning-to-match models in many applications. Furthermore, retrieval-augmented generation (RAG) [85] approaches, which combine ranking with generative AI, have introduced new ways to handle query-product matching. In this new paradigm, models like MV-LSTM and DRMMTKS, which were among the top-performing short-text matching models in our study, cannot compete with the newer architectures that better capture contextual meaning even in short product titles and descriptions.

Despite these advancements, some insights from our study remain relevant. The trade-off between accuracy and efficiency, which we examined in detail, continues to

be a critical concern. Although recently introduced Transformer-based models offer improved text representation and potentially better ranking quality, their computational expense remains a challenge for deployment in real-time, high-traffic e-commerce search, even when employing techniques such as model distillation to reduce inference cost [38]. Additionally, while these recent models can generalize better to different domains, the observation that ranking models perform differently across datasets remains an open challenge [11, 88]. While the specific models we evaluated may no longer define the state-of-the-art, the broader lessons regarding model efficiency, dataset variability, and domain-specific ranking challenges remain applicable and can inform future research in e-commerce search.

3

Understanding and Mitigating the Effect of Outliers in Fair Ranking

Traditional ranking systems of the kind discussed in the previous chapter are expected to sort items in the order of their relevance and thereby maximize their utility. In fair ranking, utility is complemented by fairness as an optimization goal. Recent work on fair ranking focuses on developing algorithms to optimize for fairness, given position-based exposure. In contrast, we identify the potential of outliers in a ranking to influence exposure and thereby negatively impact fairness. This effect is the focus of RQ2 and will be explored in this chapter. An outlier in a list of items can alter the examination probabilities, which can lead to different distributions of attention, compared to position-based exposure. We formalize outlierness in a ranking, show that outliers are present in realistic datasets, and present the results of an eye-tracking study, showing that users' scanning order and the exposure of items are influenced by the presence of outliers. We then introduce OMIT, a method for fair ranking in the presence of outliers. Given an outlier detection method, OMIT improves the fair allocation of exposure by suppressing outliers in the top-k ranking. Using an academic search dataset, we show that outlierness optimization leads to a more fair policy that shows fewer outliers in the top-k, while maintaining a reasonable trade-off between fairness and utility.

3.1 Introduction

The primary goal of a ranker as used in a search engine or recommendation system is to optimize the list to satisfy the user's needs by sorting items in their order of relevance to the query [143]. Recently, there has been a growing concern about the unfairness towards minority groups caused by this simplistic assumption [16, 21]. Several studies have proposed approaches to achieve fair ranking policies. The goal is to ensure that the protected groups receive a predefined share of visibility. Exposure-based methods [21, 103, 107, 135, 148, 149] quantify the expected amount of attention each individual or group of items receives from users in a given ranking policy, where attention is typically related to the position of the item and based on the observation that users are more likely to click on items presented at higher positions [16, 72].

This chapter was published as F. Sarvi, M. Heuss, M. Aliannejadi, S. Schelter, and M. de Rijke. Understanding and mitigating the effect of outliers in fair ranking. In *WSDM*, pages 861–869, 2022.

However, the position of the item is not the only factor that affects exposure [36]. Inter-item dependencies also play a key role [25]. For example, consider a user trying to buy a phone. When searching on an e-commerce platform, if an item in the list is on promotion and has a "Best Seller" badge, this can be distracting so that it gets more attention from the user, regardless of its position in the list; the item would stand out even more if it is the only one with this feature.

We hypothesize that inter-item dependencies have an effect on examination probability and exposure of items. We focus on the case of having an *outlier* in the ranking and aim to understand and address its effect on user behavior. We hypothesize that exposure received by an item is influenced by the existence of an outlier in the list, and assume that this effect should be considered while allocating exposure to protected groups in a fair ranking approach.

We define *outliers* as items that observably deviate from the rest. The properties and method with which we identify outliers in a set of items depend on the task. The properties are observable item features that can be presentational in nature or correspond to ranking features used to produce the ranked list. E.g., in the e-commerce search example, if only one item on a result page has a "Best Seller" tag, it is an outlier based on this presentational feature.

In this chapter, we address the following research question:

RQ2 Do outlier items exist in search logs, and how can their effect on exposure-based fair ranking algorithms be mitigated?

To begin, we perform an **exploratory analysis on the TREC Fair Ranking dataset**. We observe that a large number of outliers exist in the rankings, where we use multiple outlier detection techniques to identify outliers based on the papers' citations, as they can make an item more attractive and catchy than others.

Next, we conduct an **eye tracking study**, where we measure the attention that each item on a ranked list gets through eye tracking, so as to show that users can actually perceive outliers in rankings. We find that attention is more focused on outlier items. The scanning order and exposure received by each item may be influenced by the existence of outliers. Unlike other types of bias studied in search and recommendation [72, 113, 117, 169, 186], our eye-tracking study reveals that outlierness comes from inter-item dependencies. It affects the examination propensities for items around the outlier in a way that is less dependent on the position and based on relationships between items presented together. However, it is translated into bolded keyword matches in the title and abstracts, which can be calculated for each item separately, independently of its neighbors. Attractiveness does not alter the examination model based on position bias and only results in relatively more clicks on items when they are presented with more bolded matched keywords [186].

While allocating fair exposure to protected items or groups in a fair ranking solution, we should account for the effect of outliers. We **propose an approach to account for the existence of outliers in rankings without damaging the utility or fairness of the ranking** by mitigating outlierness, called OMIT. OMIT jointly optimizes (i) user utility, (ii) item fairness, and (iii) fewer outliers in top-k positions as a convex optimization problem that can be solved optimally through linear programming. Via its solution, we derive a stochastic ranking policy using Birkhoff-von Neumann (BvN) decomposi-

tion [23]. OMIT reduces the number of outliers at top-10 positions on the TREC 2020 dataset by 80.66%, while maintaining the NDCG@10, compared to a state-of-the-art ranking baseline.

The main contributions of this chapter are as follows: (i) we introduce, study and formalize the problem of outlierness in ranking and its effects on exposure distribution and fairness; (ii) we run an extensive eye-tracking user study in two search domains to support our hypothesis about the existence of an effect of outliers on items' exposure; (iii) inspired by our analysis, we propose OMIT, an efficient approach that mitigates the outlierness effect on fairness; (iv) we compare OMIT to competitive baselines on two TREC datasets in terms of fairness, outlierness, and utility; OMIT is able to remove outliers while balancing utility and fairness; and (v) we make the data from our eye-tracking study plus the code that implements our baselines and OMIT publicly available.

3.2 Background

Exposure and utility. Consider a single query q, that we will often leave out for notational simplicity, for which we want to rank a set of documents $\mathcal{D} = \{d_1, d_2, \ldots, d_N\}$. Suppose we are given document utilities $\mathbf{u} \in \mathbb{R}^N$, where u_i is a proxy for the relevance of document d_i for q. Let $\mathbf{v} \in \mathbb{R}^N$, be the *attention vector*, where v_j denotes how much attention a document gets at position j, and which is decreasing with the position. This vector encodes the assumed position bias, e.g., $v_j = 1/\log(1+j)$.

We require a probabilistic ranking in the form of a doubly stochastic document-position matrix $\mathbf{P} \in [0,1]^{N \times N}$ where P_{ij} denotes the probability of putting document d_i at position j. Such a matrix can be decomposed into a convex combination of permutation matrices, which allows us to sample a concrete ranking [148].

The *exposure* of a document d_i under ranking $\mathbf{P}$ denotes the expected attention that this document will get. Using the position based attention vector v, this can be modeled as a function of the ranking and position bias: $\text{Exposure}(d_i|\mathbf{P}) = \sum_{j=1}^{N} P_{ij}\, v_j$.

The *expected utility* U of a ranking $\mathbf{P}$ is the sum of the documents' utilities weighted by the exposure given to them by $\mathbf{P}$:

$$U(\mathbf{P}) = \sum_{i=1}^{N} u_i \ \text{Exposure}(d_i|\mathbf{P}) = \sum_{i=1}^{N}\sum_{j=1}^{N} u_i\, P_{ij}\, v_j = \mathbf{u}^T\mathbf{P}\mathbf{v}. \tag{3.1}$$

Without fairness considerations, a utility-maximizing ranking can be found by sorting the documents in descending order of utility.

Group fairness. Suppose now that the documents $\mathcal{D}$ can be partitioned into two disjoint sets $\mathcal{D}_{dis}$ and $\mathcal{D}_{priv}$, where documents in $\mathcal{D}_{dis}$ belong to a historically disadvantaged group (e.g., publications from not so well established institutes), and those in $\mathcal{D}_{priv}$ belong to the privileged group (e.g., publications from well-established institutes). We want to ensure a certain notion of fairness in the ranking. We want to avoid *disparate treatment* of the different groups. We use the *disparate treatment ratio* [148], which measures how unequal the exposure given to the disadvantaged group (in relation to the corresponding utility of the disadvantaged group) is compared to the corresponding

ratio of the privileged group, as:

$$\mathrm{dTR}(\mathcal{D}_{dis}, \mathcal{D}_{priv}|\mathbf{P}) = \frac{\sum_{d_i \in \mathcal{D}_{dis}} \mathrm{Exposure}(d_i|\mathbf{P}) / \sum_{d_i \in \mathcal{D}_{dis}} u_i}{\sum_{d_p \in \mathcal{D}_{priv}} \mathrm{Exposure}(d_p|\mathbf{P}) / \sum_{d_p \in \mathcal{D}_{priv}} u_p}. \tag{3.2}$$

Note that dTR is 1 if the groups are treated *fairly* and smaller than 1 if the ranking is unfair towards the disadvantaged group. We often encounter disparate treatment when only optimizing for the expected utility of a ranking [16, 21, 148]. We can find a utility maximizing ranking $\mathbf{P}$ that avoids disparate treatment by solving the following optimization problem [148]:

$$\begin{aligned} \mathbf{P} = \arg\max_{\mathbf{P}} & \mathbf{u}^\top \mathbf{P} \mathbf{v} && (\textit{expected utility}) \\ \text{such that } & \mathbb{1}^\top \mathbf{P} = \mathbb{1}^\top && (\textit{row stochasticity}) \\ & \mathbf{P}\mathbb{1} = \mathbb{1} && (\textit{column stochasticity}) \\ & 0 \leq \mathbf{P}_{ij} \leq 1 && (\textit{valid probabilities}) \\ & \mathbf{f}^\top \mathbf{P} \mathbf{v} = 0, && (\textit{dTR constraint}) \end{aligned} \tag{3.3}$$

where $\mathbb{1}$ denotes a vector and $\mathbf{f}$ is the vector constructed to encode the avoidance of disparate treatment, with

$$f_i = \frac{\mathbb{1}_{d_i \in \mathcal{D}_{dis}}}{|\mathcal{D}_{dis}| \sum_{d_s \in \mathcal{D}_{dis}} u_s} - \frac{\mathbb{1}_{d_i \in \mathcal{D}_{priv}}}{|\mathcal{D}_{priv}| \sum_{d_p \in \mathcal{D}_{priv}} u_p}, \tag{3.4}$$

where $\mathbb{1}_{d_i \in \mathcal{D}_{dis}} = 1$ if document d_i is in the disadvantaged group and 0 otherwise (and analogously for $\mathbb{1}_{d_i \in \mathcal{D}_{priv}}$) [148].

Degrees of outlierness. Outliers are items that deviate from the rest of the data [69]. They can be interesting observations or suspicious anomalies. Either way, they are considered noise that can affect the statistical analysis. We describe three outlier detection methods that we will use later in this chapter. Let $x = \{x_1, \ldots, x_n \,|\, x_i \in \mathbb{R}\}$ be a set of values for which we want to identify outliers.

Median Absolute Deviation (MAD). Although it is common practice to use Z-scores to identify possible outliers, this can be misleading (particularly for small sample sizes) due to the fact that the maximum Z-score is at most $(n-1)/\sqrt{n}$. Iglewicz and Hoaglin [64] recommend using the modified Z-score: $M_i = 0.6745(x_i - \tilde{x})/MAD$, where MAD is the median absolute deviation and $\tilde{x}$ is the median of x. These authors recommend that modified Z-scores with an absolute value of greater than 3.5 be labeled as potential outliers.

Median K-Nearest Neighbor (MedKNN). This model [10] uses the K-Nearest Neighbor algorithm to define a distance-based outlier detection method. For each point x_i we have a value $w_k(x_i)$ as the weight calculated from the k nearest neighbors; outliers are the points with the largest values of w_k. We use Median K-Nearest Neighbor, which computes $w_k(x_i)$ as the *median* distance of x_i to its k neighbors. In order to find the k nearest neighbors, the method linearizes the search space and uses *Hilbert space-filling curve* for fast and efficient search; the method scales linearly in the dimensionality and

the size of the dataset.

Copula-Based Outlier Detection (COPOD). COPOD [89] is a novel outlier detection method based on estimating tail probabilities using empirical copula. COPOD uses empirical cumulative distribution functions (ECDFs) to compute tail probabilities. These tail probabilities estimate the probability of observing a point at least as extreme as x_i for each data point x_i. If x_i is an outlier, the probability of observing a point as extreme as x_i should be small and it means that this point has a rare occurrence. This method is deterministic and efficient, and scalable to high dimensionality data.

3.3 Outliers in Ranking

A common assumption is that exposure is a function of position [21, 103, 107, 148, 167]. We argue that this assumption holds only if ranked items can be deemed similar, meaning that no item is perceived as an outlier. Below we introduce outliers in the context of ranking. We then determine that outliers are present in rankings in realistic datasets. We also report on an eye-tracking study that shows that the presence of outliers in ranked lists impacts user behavior.

A definition of outliers in ranking. For a ranked list, we define outliers as items that stand out among the window of items that the user can see at once, drawing the user's attention. Outlier items often have (visible) characteristics that distinguish them from their neighbors. E.g., consider Figure 3.1, which shows a result page for the query "smart phone". The result page view consists of 6 products, each presented with characteristics such as title, image, and price. The item at position three deviates from the other items in terms of several visual characteristics; it has more details, some promotive tags, and bold keywords. Other features, such as more positive reviews, or a higher price, may also influence the user's perceived relevancy. In this example, the third item can be considered as an outlier according to such visual characteristics.

Formally, we define outliers in ranking as follows. Consider a ranked list of N items in $\mathcal{D} = \{d_1, d_2, \ldots, d_N\}$, that has been produced in response to a query. We call an observable characteristic of an item d in a ranked list an *observable item feature*. These features can be purely presentational in nature, like the bold keywords in Figure 3.1, or correspond to ranking features used by the search engine to produce the ranked list, e.g., the average user rating.

Definition 1 (Degree of outlierness)**.** Let g be an observable item feature, and M be one of the outlier detection methods discussed in Section 3.2. The *degree of outlierness* of an item d_i in the ranked list $[d_1, \ldots, d_N]$ is the value calculated by M for $g(d_i)$ in the context of $\{g(d_1), \ldots g(d_N)\}$, that determines how much $g(d_i)$ differs from the other elements of the set. We write $\mathrm{M}(g(d_i)|\mathcal{D})))$ for this value.

Definition 2 (Outliers in ranking)**.** We say that according to M, item d_i is an *outlier in the ranked list* $[d_1, \ldots, d_N]$ *for feature* g, if M identifies $g(d_i)$ as an outlier in the set $\{g(d_1), \ldots, g(d_N)\}$.

Note that detecting an item as an outlier in a ranking depends on the context in which we see the item. Throughout this chapter, we consider the full ranked list of items as

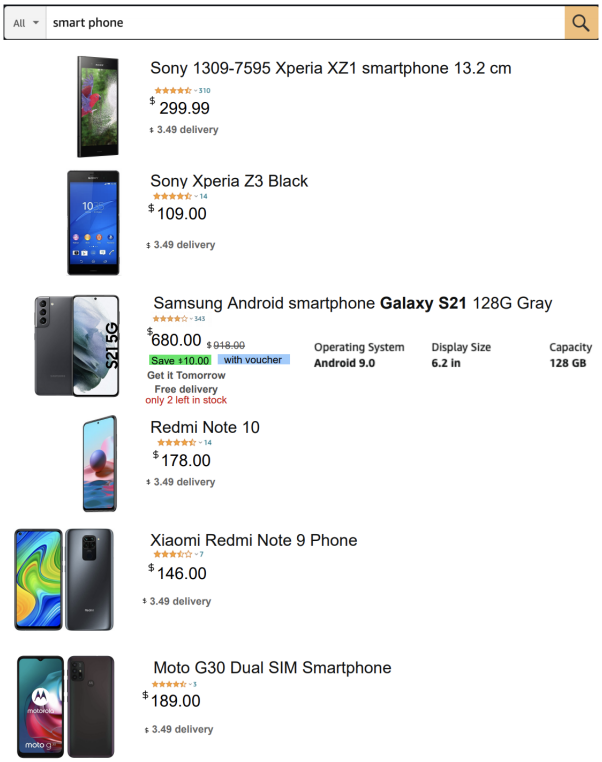


Figure 3.1: E-commerce example used in our eye-tracking user study. A result page with one outlier at position 3, identified by more descriptive fields, higher price, and colored tags.

the context in which we detect outliers. In Section 3.6 we study varying sizes for the context.

Moreover, it is possible to use multiple observable features to detect the outliers. For example, we can consider image size as g_1 and price as g_2, and then use any combination of these two feature values to present item d_i.

Below, when we refer to an item d being an outlier in a given ranked list, we assume that it is clear from the context what outlier detection method M and observable item feature g are being used.

Do outliers in ranking exist? To determine whether outliers are present in rankings in datasets, we take a retrieval test collection, compute feature values for one of the (potentially observable) rankings features appropriate for the collection, and determine whether there are outliers among the top-20 documents returned for the test queries (using ListNet as the ranker, see Section 3.5). For the experiments in Section 3.6 we use the academic search dataset provided by the TREC Fair Ranking track.[1] It contains information about papers and authors extracted from the Semantic Scholar

[1] https://fair-trec.github.io/

Table 3.1: Descriptive statistics of the TREC2020 Fair Ranking Track dataset.

	Training	**Test**
#queries	200	200
#unique authors	16,499	17,571
#unique papers	4,649	4,693
% of clicks in sessions	0.169	0.170

Open Corpus.[2] It comes with queries and relevance judgments; see Table 3.1 for some descriptive statistics.

We used the number of citations as observable feature g for this dataset as they can make an item more attractive than others (when reported). In the remainder of the section, we report the analysis only on the TREC 2020 data, as we observed similar trends in both datasets. Figure 3.2 shows the mean, maximum, and minimum of papers' citation counts for all search sessions in the data. There is a high variance between mean and maximum citations, which implies that the data is outlier-prone based on this feature. We plot outlier counts for each position in the top-20. Figure 3.3 depicts the number of relevant and non-relevant outliers detected by the outlier detection methods introduced in Section 3.2 at different positions. The stacked bars show that in spite of the attractiveness of outliers, most of these items are irrelevant, judging by the click data. In total, 88.5%, 89.8%, and 90.1% of the outliers are irrelevant when MedKNN, COPOD, and MAD are employed as the outlier detection method, respectively. The average percentage of irrelevant documents in the top 20 positions in the dataset is 83.3%. This suggests that by pushing these items to lower positions we can improve the degree of outlierness without jeopardizing utility.

Do users perceive outliers in the ranking? Outliers are present in realistic datasets, but do they impact user behavior? Prior studies stress the importance of relationships between ranked items [123], but it is unknown how an outlier in a ranking affects the examination probability. To address this gap, we conduct an eye-tracking study. We ask participants to interact with search engine result pages, as they normally would, and find the items that they prefer and think are relevant. We track their eye movements via an online webcam-based eye tracking service.[3] We use two scenarios, e-commerce, and scholarly search. We focus on a list view; in both scenarios, participants are able to see all items in one page. For each scenario, we include two result pages, one without an outlier item and one with (as in Figure 3.1). In the absence of outliers we expect participants to follow the position bias examination assumption [72]; in the presence of outliers, we expect that users' attention is diverted towards them.

We recruit 40 university students and staff for both scenarios. In the instructions, we describe the overall goal of the research and ask participants to read the instructions carefully. We describe what webcam-based eye tracking is and that the eye-tracking service will ask them for access to their webcam. We instruct participants to first read and understand the query and then start scanning the result page as if they submitted the query themselves.

[2] `http://api.semanticscholar.org/corpus/`

[3] RealEye, `https://realeye.io`.

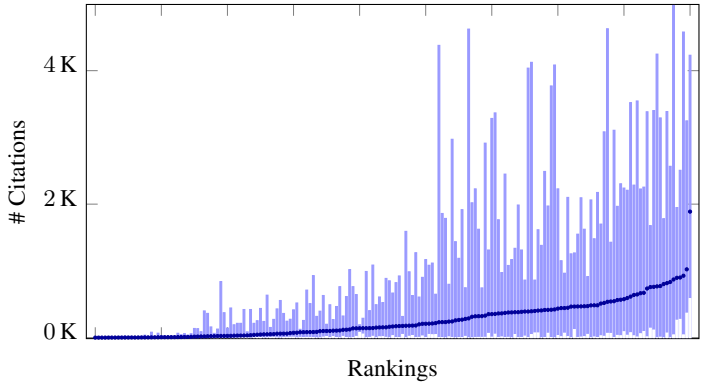


Figure 3.2: Distribution of the number of citations of top-20 papers returned for test queries in the TREC 2020 Fair Ranking track.

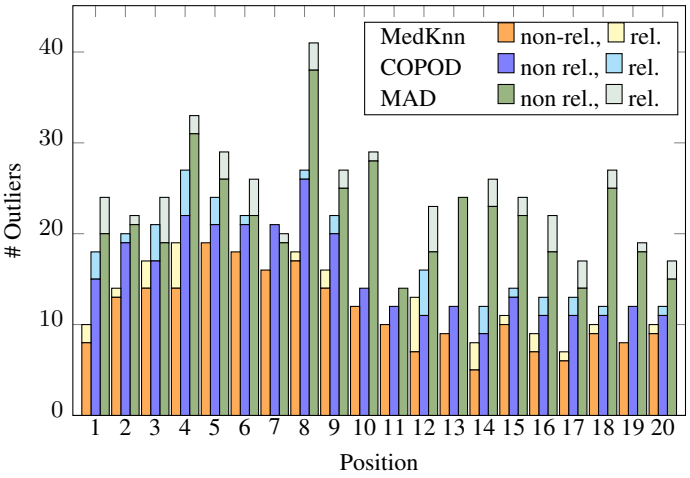
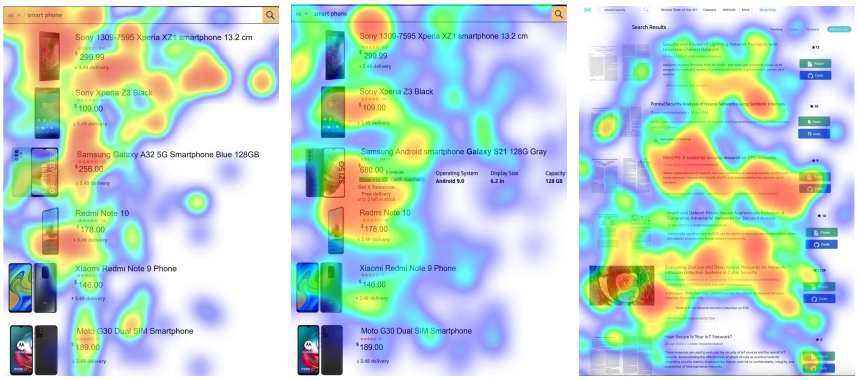


Figure 3.3: Number of outliers at each position w.r.t. different outlier detection methods, considering the number of citations as the observable feature. Each stacked bar shows the number of irrelevant and relevant outliers.

For reporting, we consider four eye-tracking measures based on participants' eye fixations: (i) fixation count (the number of fixations within an area; more fixation means more visual attention); (ii) time spent (shows the amount of time that participants spent on average looking at an area); (iii) Time To First Fixation (TTFF; the amount of time that it takes participants on average to look at one area for the first time); and (iv) revisit count (indicates how many times on average participants looked back at the area) [49].

Outliers in e-commerce. For this scenario, we mimic the Amazon Marketplace[4] search result page. Figure 3.1 depicts our example ranked list with an outlier. The third item in the list stands out from other items for different reasons, including price and

[4] `https://www.amazon.com`



(a) (b) (c)

Figure 3.4: Examples used in the eye-tracking study. (a) Heatmap for a result list without outlier for the query "smart phone". Items at the top of the result list receive more attention, following the position bias assumption. (b) Heatmap for a similar result list (the same list as in Figure 3.1) but with one outlier, at position 3. Participants exhibit increased attention towards and around the outlier item. (c) Heatmap for a scholarly search example, with an outlier (at position 4).

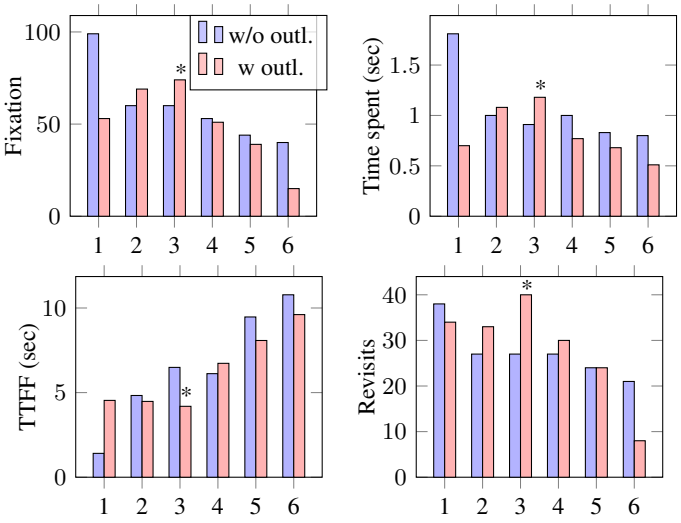
sales-related tags (e.g., being on sale), as well as other information that is available for this item. Comparing Figure 3.4a and 3.4b, we see that in the presence of an outlier, items at the top of the list receive less attention, contradicting the position bias assumption.

Figure 3.5(a) reports the eye-tracking measures for the e-commerce scenario. We highlight the outlier in each list with an asterisk. In the no-outlier condition, participants exhibit linear behavior in terms of scanning the items. The highest number of fixations, time spent, revisits belongs to the items on the top of the list, and it decreases as we go down the list. TTFF demonstrates a linear behavior of the first fixation time, that is, most of the participants started scanning the results from top to bottom. The ranked list with an outlier exhibits very different measurements. Attention is more focused on the outlier item. Also, we see that from the TTFF values, the average time for the first fixation is the lowest for this item, suggesting that the scanning order and exposure are influenced by the existence of the outlier. This is also evident by comparing the heatmaps in Figure 3.4a and 3.4b.

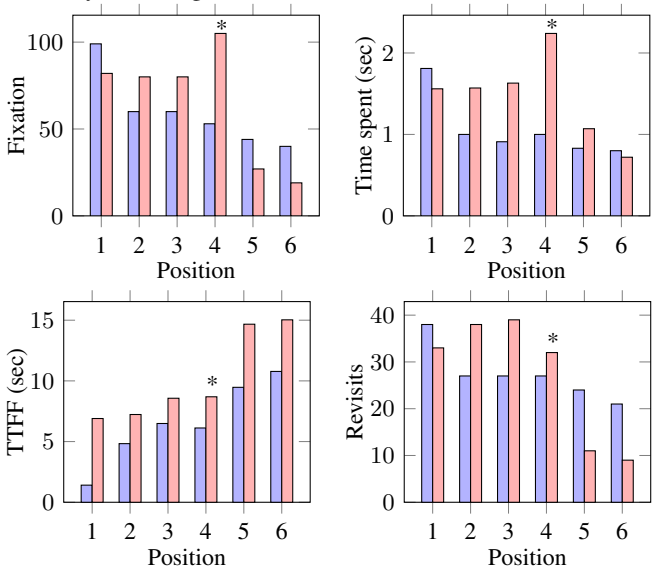
Outliers in scholarly search. In the second scenario, we study the effect of outliers on scholarly search result pages. To this end, we mimick the result page from PapersWithCode.[5]

To save space, we omit the eye fixation heatmap for the result page without outliers; it shows the familiar F-shape. Figure 3.4c shows the eye fixation heatmap for the result page *with* an outlier item; the fourth item has a different thumbnail and a large number of GitHub stars, making this item stand out in the list. Similar to the e-commerce scenario, the eye fixations are very different from the F-shape typical for the no-outlier

[5] `https://paperswithcode.com`



(a) Eye tracking measurements for e-commerce search.



(b) Eye tracking measurements for scholarly search.

Figure 3.5: Eye tracking measurements for each position, based on participants' eye fixations. The positions where outliers were shown are marked with an asterisk.

case. Fixations and time spent are the highest for the outlier item, suggesting that it draws lots of attention, and contradicts the cascade examination assumption.

However, different from the e-commerce case, we do not observe as big a difference in the TTFF values between the conditions with and without outliers, suggesting that the order of item scans is not affected as much as in the e-commerce example.

3.4 Mitigating Outlierness in Fair Learning to Rank

We now present a ranking algorithm for mitigating outlierness for fairness in ranking, called OMIT, that simultaneously accounts for item fairness and outlierness effect requirements. Based on the observations reported in the previous section we know that outliers can influence exposure and examination order, in a way that can be considered as a type of bias. We take a first step towards mitigating the outlierness phenomenon by proposing a remarkably simple, yet effective solution that removes outliers from the top positions where the distribution of exposure is most critical. Our solution aims at decreasing outlierness in the top-k positions, while retaining the ranking's utility and fairness with position-based assumptions.

OMIT is based on the linear programming method described in Section 3.2. In addition to optimizing for user utility while staying within the fairness constraints, our goal is to reduce the number of outliers in the top-k of all rankings. OMIT has two steps. In the first, we search for the marginal rank probability matrix that satisfies item group fairness by solving a linear program that optimizes both for user utility and fewer outliers in the top-k items with linear constraints for fairness. In the second, we derive a stochastic ranking policy from the solution of the linear programming method using the Birkhoff-von Neumann decomposition [23]; cf. also [148].

Step 1: Computing MRP matrix. Let $\mathcal{D}_q = \{d_1, \ldots, d_N\}$ be a set of items that we aim to rank for a given query q. Each ranking $\sigma(\cdot|q)$ corresponds to some permutation matrix P_σ that re-orders the elements of $\mathcal{D}_q$.[6] As discussed in Section 3.3, to determine which items are outliers, we use the domain specific observable item features $g(d_1), \ldots, g(d_N)$ that potentially impact the user's perception of an item. Using these characteristics as features, one can use any outlier detection method to determine which items should be considered outliers. The majority of outlier detection methods, including the ones we use (Section 3.3), find outliers by calculating scores that indicate the degree of outlierness. This results in a list of outlierness values $\mathrm{M}(g(d_1)|\mathcal{D}), \ldots, \mathrm{M}(g(d_n)|\mathcal{D})$, where $n \leq N$ is the size of the outlierness context that the algorithm takes into account while detecting the outliers. We define a vector $\mathbf{o}_\mathcal{D}$, that contains, for each document, the information whether it is an outlier with respect to the full ranked list:

$$\mathbf{o}_\mathcal{D} = \{o_i\}_i \quad \text{with } o_i = \begin{cases} \mathrm{M}(g(d_i)|\mathcal{D}), & \text{if } d_i \text{ is outlier in } \mathcal{D} \\ 0, & \text{else.} \end{cases} \tag{3.5}$$

We use $\mathbf{o}$ and $\mathbf{o}_\mathcal{D}$ interchangeably. Note that we are considering outliers in the context of the full list, $n = N$, i.e., the outlier detection algorithm takes the whole ranked list as input. We assume that items that are perceived as outliers in this context are likely to be perceived as outliers when seen in the smaller context of the top-k items. Below, we show that this heuristic works well in practice.

OMIT works by pushing outliers away from the top-k. Let P_σ be the permutation matrix corresponding to a ranking σ. The amount of outlierness that a ranking σ places

[6] For simplicity, we interchangeably use $\sigma(\cdot)$ and $\sigma(\cdot|q)$, as well as $\mathcal{D}$ and $\mathcal{D}_q$.

in the top-k are equal to

$$\text{Outlierness}_k^{\mathcal{D}}(\sigma|\,\mathrm{M}) = \sum_{i=0}^{k}(\mathbf{o}^T P_\sigma)_i = \mathbf{o}^T P_\sigma \mathbf{h}, \tag{3.6}$$

where $\mathbf{h} = (1, \ldots, 1, 0, \ldots, 0)$ is a vector containing 1 for the first k positions and 0 for all positions after that. Similarly, the expected outlierness, that is placed in the top-k by $\mathbf{P}$ is given by

$$\text{Outlierness}_k^{\mathcal{D}}(\mathbf{P}|\,\mathrm{M}) = \mathbf{o}^T \mathbf{P} \mathbf{h}. \tag{3.7}$$

While optimizing for user utility we can use the expected outlierness to add an objective that will function as a regularization term, penalizing ranking policies with outliers in the top-k. We extend the linear programming approach described in Section 3.2 to solve:

$$\begin{aligned} \mathbf{P} = \arg\max_{\mathbf{P}} & \mathbf{u}^T \mathbf{P} \mathbf{v} - \mathbf{o}^T \mathbf{P} \mathbf{h} \\ \text{such that } & \mathbf{P} \text{ is doubly-stochastic} \\ & \mathbf{P} \text{ is fair.} \end{aligned} \tag{3.8}$$

For item fairness, we adopt the disparate treatment constraints as described in Section 3.2. Both terms of the optimization objective, and the constraints for fairness and finding a doubly stochastic matrix are linear in N^2 variables. Hence, the resulting linear program can be solved efficiently using standard optimization algorithms [148].

Step 2: Constructing a stochastic ranking policy. The solution to the linear program $\mathbf{P}$ is a matrix indicating the probability of showing each item at any position. To generate actual rankings, we need to derive a stochastic ranking policy π from $\mathbf{P}$ and sample rankings σ to present to users. We follow [148] and use Birkhoff-von Neumann decomposition to compute π, which decomposes the doubly stochastic matrix $\mathbf{P}$ into the convex sum of permutation matrices $P = \theta_1 P_{\sigma_1} + \cdots + \theta_n P_{\sigma_n}$, with $0 \leq \theta_i \leq 1, \sum_i \theta_i = 1$ [23]. This results in at most $(N-1)^2 + 1$ rankings σ_i [100], corresponding to P_{σ_i}, that are shown to the user with probability θ_i, respectively.[7]

OMIT model summary. Algorithm 1 presents an overview of OMIT. OMIT takes as input the initial ranking D_q (optimized for utility), outlier detection method M, outlierness context size n, and the number of top items that we aim to remove outliers k. In line 1, $\mathbf{o}_{\mathcal{D}}$ is created for a given outlier detection technique and outlierness context size, followed by line 2 where we create the $\mathbf{h}$ list that takes into account the top of the list that we aim to mitigate outlierness. In line 3, we solve the linear program that jointly solves the fairness and outlierness problem and pass the stochastic ranking in line 4 to the BvN decomposition algorithm. Finally, we return the output of the BvN method as the output.

3.5 Experimental Setup

To answer RQ2, we investigate the following research questions:

[7] We used the implementation from `https://github.com/jfinkels/birkhoff`.

Algorithm 1 Outlier mitigation for fair ranking (OMIT)

Input: $\mathcal{D}_q$, M, n, k
Output: π
1: Create $\mathbf{o}_{\mathcal{D}}$ as Eq. (3.5) using $\mathcal{D}_q$ and M
2: $\mathbf{h} \leftarrow (h_1, \ldots, h_n)$ such that $h_i = 1$ if $i \leq k$ else 0
3: $\mathbf{P} \leftarrow \arg\max_{\mathbf{P}} \mathbf{u}^T\mathbf{P}\mathbf{v} - \mathbf{o}^T\mathbf{P}\mathbf{h}$ such that $\mathbf{P}$ is doubly-stochastic and fair (Eq. (3.8))
4: $\pi \leftarrow$ BvN-Decomposition($\mathbf{P}$)
5: Return π

RQ2.1 How do different outlier detection methods affect OMIT's performance in terms of utility, fairness, and outlierness?

RQ2.2 How does our OMIT trade-off between utility, fairness, and outlierness, compared to baselines?

RQ2.3 We adopt the constraints proposed in [148] (called FOE) to optimize a ranked list for fairness and utility through linear programming, as described in Section 3.4. Given that OMIT adds additional constraints, is the overall linear program more effective when we treat the doubly stochastic matrix constraints as hard or soft constraints?

RQ2.4 How does changing the context of detecting outliers affect OMIT's outlierness improvement and utility?

RQ2.5 How does changing k affect OMIT's outlierness improvement and utility in the top-k positions?

Data. We use data from the TREC Fair Ranking 2019 and 2020 track (see Section 3.3). We make the group definitions over the two datasets consistent. As for the TREC 2019 data, we bin the original article groups into two classes. For the TREC 2020 data, we adopt the group definitions from the original data, that is, documents are assigned to two groups based on their authors' h-index. Moreover, we follow the TREC setup to generate query sequences, leading to multiple occurrences of the same query, using the provided frequencies. Specifically, we evaluate on a query sequence of size $10,000$, including all the queries in the evaluation data.

Evaluation metrics. We evaluate methods for fair learning to rank in the presence of outliers in terms of utility, item fairness, and outlierness. For utility and fairness, we use NDCG and dTR (see Eq. (3.2)), respectively, as our metrics and report their expected values.

To measure the expected outlierness of the policy $\mathbf{P}$ up to position n in the ranking, we use $\text{Outlierness}_n^{\mathcal{D}}(\mathbf{P}|\,\text{M})$ as defined in Eq. (3.7). Similarly we define the expected number of outliers up to position n in ranking for policy $\mathbf{P}$ as

$$\#\text{Outliers}_n^{\mathcal{D}}(\mathbf{P}|\,\text{M}) = \mathbf{o}_b^T\mathbf{P}\mathbf{h}, \tag{3.9}$$

where $\mathbf{o}_b^T = \mathbb{1}_{>0}(\mathbf{o}^T)$ is the binarized version of $\mathbf{o}^T$ where each outlier item is assigned 1, and all the rest are assigned 0.

Compared methods. To evaluate OMIT, we build several baseline methods, combining different options for each component of our model (initial ranking, fairness of exposure, outlier mitigation):

- **Initial ranking:** The initial ranking of all compared methods is generated using ListNet [29]. ListNet is a learning to rank model, optimizing for utility. We use it in our experiments to create initial ranked lists, $\mathcal{D}_q$, using the click data provided in the training set, with 30 maximum epoch, and a validation ratio of 0.3.
- **Fairness of exposure**: We use two variants of FOE [148] based on hard vs. soft doubly stochastic matrix constraints, and call them FOE^H and FOE^S, respectively.[8]
- **Outlier mitigation**: We employ two outlier mitigation techniques, namely, RO and OMIT. RO removes all the outlier items detected by M from the ranking, while OMIT is our proposed outlier mitigation method as described in Section 3.4.

We specify methods as combinations of the three components mentioned above. E.g., "ListNet + FOE^H + OMIT" uses the initial ranked list produced by ListNet, applying FOE fairness post-processing with hard constraints and the OMIT outlier mitigation model.

3.6 Empirical Results

Effect of outlier detection method. We address RQ1.1 by changing the outlier detection method, while keeping the other parts of the model fixed. Table 3.2 reports the results of using three different outlier detection methods in OMIT. For comparison, we also report the results of ListNet without outlier mitigation (row **a**) and report the relative improvements. All three outlier detection methods effectively reduce outlierness compared to the baselines. COPOD achieves the best results by reducing outlierness by 80.3% and 80.6% in terms of Outlierness@10 on the TREC 2019 and 2020 data, respectively. In terms of dTR, COPOD outperforms MAD and MedKNN on both datasets where it increases dTR by 21.9% on TREC 2020. We see no significant difference in the utility achieved by the methods. Given that COPOD is parameter-free and scalable, we prefer it over the other two methods. For the remaining experiments, we choose COPOD as the outlier detection method.

Utility, fairness, and outlierness trade-offs. To answer RQ1.2, we turn to Table 3.2, which shows the results for OMIT and the baseline methods in terms of utility, fairness, and outlierness.

Although ListNet is purely optimized for utility, it does not achieve the highest NDCG in all cases. This suggests that optimizing for fairness and outlierness could even improve the utility.

As shown in Figure 3.3 there is a high density of outliers among top positions that are mostly irrelevant. Therefore, when OMIT pushes these items to lower positions, it improves outlierness and utility measures simultaneously. Moreover, we see that

[8] We used the implementation from `https://github.com/MilkaLichtblau/BA_Laura`.

Table 3.2: Comparing loss in fairness and utility, with gains in outlierness for different outlier detection methods on the TREC 2019 and 2020 Fair Ranking data. Models used: (a) ListNet and (b) ListNet + FOE^S + OMIT. Δ values denote the percentage of relative improvement compared to (a). * refers to statistically significant improvements compared to (a) using a two-tailed paired t-test ($p < 0.05$).

			NDCG ↑		Fairness ↑	# Outliers ↓		Outlierness ↓	
	Model	Outl.	@5	@10	dTR	@10	$\Delta(\%)$	@10	$\Delta(\%)$
TREC 2019	(a)	COPOD	0.671	0.757	0.935	1.260	–	0.873	–
		MedKNN	0.671	0.757	0.935	0.507	–	0.432	–
		MAD	0.671	0.757	0.935	0.811	–	0.599	–
	(b)	COPOD	0.667	0.753	**0.977***	0.208*	**83.49**	0.172*	**80.29**
		MedKNN	**0.671**	0.756	0.936	0.102*	79.88	0.094*	78.24
		MAD	**0.671**	**0.757**	0.957*	0.290*	64.24	0.205*	65.77
TREC 2020	(a)	COPOD	0.240	0.356	0.790	1.043	–	0.755	–
		MedKNN	0.240	0.356	0.790	0.783	–	0.602	–
		MAD	0.240	0.356	0.790	1.456	–	0.779	–
	(b)	COPOD	0.240	0.366*	**0.963***	0.178*	**82.93**	0.146*	**80.66**
		MedKNN	0.239	0.361	0.740	0.160*	79.56	0.133*	77.90
		MAD	**0.242**	**0.372***	0.709	0.430*	70.46	0.202*	74.10

mitigating outlierness does not cause any significant effect on dTR, showing that OMIT is capable of retaining position-based item fairness.

Table 3.3 shows that OMIT effectively decreases the number of outliers in top-10 positions by at most 83.49% (and 82.93%) when used with FOE^S (row **g** in the table) on the TREC 2019 (and 2020) data. For the outlierness metrics, these values are 80.29% and 80.66%. Referring back to our data analysis, we observed a high density of non-relevant outlier items at the top of the list, indicating the high possibility of user distraction towards these non-relevant items, as suggested by our eye-tracking study.

Hard vs. soft constraint. We turn to RQ1.3 and experiment with two variants of the FOE model, which differ in the constraint for generating a doubly stochastic matrix (see Eq. (3.8)). This constraint is important since the BvN algorithm can guarantee to generate valid permutations only if the input is doubly stochastic. We observed that forcing the convex optimization to output such matrices can make the constraints too hard to satisfy even when only optimizing for fairness and utility. Hence, the algorithm cannot find an optimum solution for many of the queries. For example, FOE^H cannot find solutions for 47%, and 46% of the queries on TREC 2019 and 2020 data, respectively. We return the original ranking as the output when FOE^H does not find an optimum solution. To fix this problem we implemented the constraint for doubly stochastic matrix as a soft constraint and we check for validity of the permutation matrices generated by the decomposition algorithm. Table 3.3 demonstrates the effectiveness of the soft variant of FOE (row **g** vs. **f**).

Effect of n. To answer RQ1.4, we examine the effect of the n parameter, which determines the outlierness context size. We change n from 10 to 40 items while keeping other parameters fixed, and observing the behavior of the model. This is important since

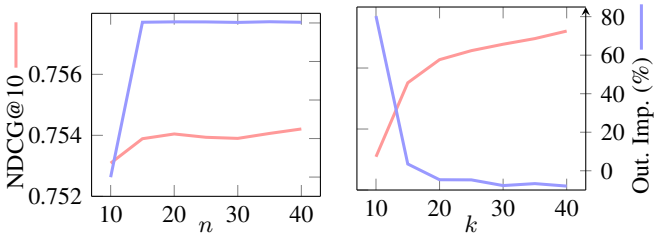
Table 3.3: Comparing loss in fairness and utility, with gains in outlierness for COPOD on the TREC 2019 and TREC 2020 Fair Ranking data. Models used: (a) ListNet; (b) ListNet + FOE^H; (c) ListNet + FOE^S; (d) ListNet + FOE^H + RO; (e) ListNet + FOE^S + RO; (f) ListNet + FOE^H + OMIT; (g) ListNet + FOE^S + OMIT. Δ values denote the percentage of relative improvement compared to (a). Other conventions are the same as in Table 3.2.

		NDCG ↑		Fairness ↑	# Outliers ↓		Outlierness ↓	
	Model	@5	@10	dTR	@10	$\Delta(\%)$	@10	$\Delta(\%)$
TREC 2019	(a)	0.671	0.757	0.935	1.260	–	0.873	–
	(b)	0.670	0.756	0.935	1.235	–	0.870	–
	(c)	**0.673**	**0.758**	0.945	1.225	–	0.852	–
	(d)	0.663	0.697	0.961*	1.114	11.58	0.861	1.37
	(e)	0.667	0.700	**0.990***	1.072*	14.92	0.834	4.47
	(f)	0.672	0.757	0.951	1.080*	14.28	0.753*	13.74
	(g)	0.667	0.753	0.977*	0.208*	**83.49**	0.172*	**80.29**
TREC 2020	(a)	0.240	0.356	0.790	1.043	–	0.755	–
	(b)	0.237	0.355	0.301	1.073	–	0.780	–
	(c)	0.241	0.357	0.766	1.046	–	0.758	–
	(d)	**0.242**	0.362	0.313	1.143	−9.58	0.811	−7.41
	(e)	**0.242**	0.362	0.840*	1.148	−10.06	0.817	−8.21
	(f)	0.237	0.359	0.314	0.885*	15.15	0.645*	14.56
	(g)	0.240	**0.366***	**0.963***	0.178*	**82.93**	0.146*	**80.66**

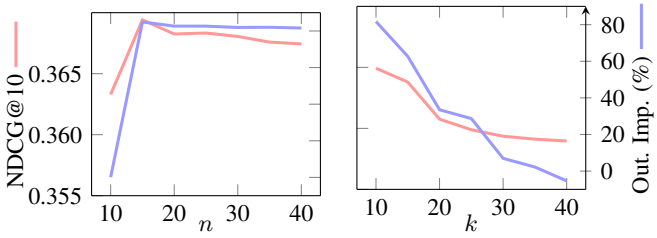
the outlierness of an item depends on its context, e.g., an item can be considered as an outlier in the top-10 items, but may not be an outlier in the top-20 items. The two left plots in Figure 3.6 show the outlierness improvements (compared to ListNet) and utility in terms of NDCG@10 for different values of n on the TREC 2019 and 2020 data. We see that Outlierness@10 improves for larger values of n, suggesting that determining outlierness of items in a bigger pool of items is more accurate and allows OMIT to mitigate the outliers in the top-10 positions more effectively.

Effect of k. Finally, we answer RQ1.5. We study the effect of changing k when optimizing for mitigating outlierness in top-k positions. We are interested in finding out how utility and outlierness are influenced when optimizing for mitigating outlierness for a longer list of top ranks ranging from 10 to 40. This mimics the cases where more items can be shown to a user, hence outliers in longer lists can be observed by the user. The left plots in Figure 3.6 depict the results for both datasets. We observe that outlierness improvement drops for greater values of k since it is more challenging for the algorithm to push all outliers to lower positions.

Figure 3.3 shows that most outliers are located at top positions, so greater values of k do not necessarily translate to more outliers. The changes in utility scores of the lists are marginal, with a 0.5% increase and 1.6% decrease for larger values of k for the TREC 2019 and 2020 data, respectively. The difference in utility scores of the two datasets is due to the fact that there are more relevant outliers in TREC 2019 than in TREC 2020.



(a) TREC Fair Ranking Track 2019 data.



(b) TREC Fair Ranking Track 2020 data.

Figure 3.6: NDCG@10 and outlierness@10 improvement percentage for different values of k and different sizes of the window in which we detect the outliers.

3.7 Related Work

Bias in implicit feedback. Users' implicit feedback, such as clicks, can be a great source of relevance judgment that has been shown to help improve search quality [2]. These signals may be misleading due to different types of bias, which causes the probability of a click to differ from the probability of relevance. Recent work on discovering and correcting for different types of bias in logged click data concerns position bias [72, 74], presentation bias [186], selection bias [118], trust bias [2, 161], popularity bias [1], and recency bias [33]. Inter-item dependencies affect the perceived relevance of items [36]. We introduce a phenomenon that is anchored in inter-item dependencies and may result in biased clicks. Our work differs from previously discussed types of bias by considering inter-item relationships. Showing items as outliers can make them more attractive to users, influencing their perceived relevance. Presentation bias [186] concerns a related effect; bold keywords in titles make some items appear more attractive. However, this definition of attractiveness is independent of other items in the list. Metrikov et al. [104] show that click-through rates can be manipulated by adding more images to top positions next to the ad slots on the search result page; they did not study the effect on item exposure or biased clicks. We focus on the effect of outliers, as an inter-item dependency on the examination probability and item exposure, and introduce it as a potential source of bias in click data.

Fair ranking. Following [188] we distinguish two ways of measuring the fairness of the rankings. Work on *probability-based* fairness determines the probability that a ranking is the result of a fair process [14, 30, 31, 51, 151, 183, 187]. *Exposure-based* methods

determine the expected exposure for each item in the ranking and aim to ensure that this exposure is fairly distributed [21, 42, 103, 107, 135, 148, 149, 182]. Our work belongs to the second category. To estimate the expected exposure of each item or group, we need to take into account different types of bias that the user might have when observing system output. Previous work has mainly focused on position bias [21, 148, 182]. We emphasize the role of inter-item dependencies in the exposure that an item receives, which can be a source of unfairness when not considered in computing the expected exposure. We extend the re-ranking approach introduced in [148], to not only produce fair rankings but also avoid showing outliers in the top-k. An important effort to develop a benchmark for the evaluation of retrieval systems in terms of fairness is *TREC Fair Ranking track* (see footnote 1). We expand the use of the track resources to include the study of outliers in fair ranking.

3.8 Conclusion & Future Work

We introduced and studied a phenomenon related to fair ranking, that is, outlierness. We analyzed data from the TREC Fair Ranking track and found a significant number of outliers in the rankings. We hypothesized that the presence of outliers in a ranking impacts the way users scan the result page. We confirmed this hypothesis with an eye-tracking study in two scenarios: e-commerce and scholarly search. We proposed OMIT, an approach to mitigate the existence and effect of outliers in a ranked list. With OMIT, we introduced a ranking constraint based on the outlierness of items in a ranking and combined it with fairness constraints. We formulated the problem of outlier mitigation as a linear programming problem and produced stochastic rankings, given an initial ranking. Using OMIT one can reduce outliers in rankings without compromising user utility or position-based item fairness. We analyzed the effects of different outlier detection approaches and compared their results. Our experiments also showed that there is a trade-off between the depth of outlier detection and user utility. Now that we have established the impact of outliers in rankings, future work on fair ranking should consider the presence of outliers by default.

In the next chapter, we will estimate the impact of this inter-item dependency on user examination probabilities. We will introduce outlier bias and propose a method to estimate and account for this type of bias by answering RQ3.

Table 3.4: Notation used in Chapter 3.

Notation	Gloss
$\mathbb{1}$	ones vector
d	item to be ranked
$\{d_1, ...d_N\}$	set of documents
$[d_1, ...d_N]$	ranked list of documents
$\mathcal{D}$	set of items to be ranked for a single query
$\mathcal{D}_q$	set of items to be ranked for query q
$\mathcal{D}_{dis}$	set of items that belong to privileged group (for a single query)
$\mathcal{D}_{priv}$	set of items to be ranked for query q
DTR	Disparate Treatment Ratio
$\text{Exposure}(d_i\|\mathbf{P})$	exposure given to item group G_i
$\mathbf{f}$	fairness vector
$\mathbf{h}$	vector with ones for top-k and zero else
g	observable item feature
M	outlier detection method
$\text{M}(g(\cdot)\|\mathcal{D})$	degree of outlierness in $\mathcal{D}$
$\mathbf{o}$	outlier vector for outlier objective
o_i	entries of vector $\mathbf{o}$
MAD	mean absolute deviation
$\tilde{x}$	median of x
π	ranking policy
$\mathbf{P}$	Marginal Rank Probability matrix (MRP)
$P_{i,j}$	entries in MRP
$\mathbf{P}_\sigma$	permutation matrix corresponding to σ
q	query
σ	ranking
$\mathbf{u}$	document utilities $\{u_i\}$ (per item)
U	expected utility
$\mathbf{v}$	attention vector $\{v_i\}$ (per position)

4

On the Impact of Outlier Bias on User Clicks

User interaction data is an important source of supervision in counterfactual learning to rank (CLTR). Such data suffers from presentation bias. Much work in unbiased learning to rank (ULTR) focuses on position bias, i.e., items at higher ranks are more likely to be examined and clicked. In Chapter 3 we have shown that inter-item dependencies also influence examination probabilities, with *outlier* items in a ranking as an important example. Outliers are defined as items that observably deviate from the rest and therefore stand out in the ranking. In this chapter we identify and introduce the bias brought about by outlier items: users tend to click more on outlier items and their close neighbors.

To address RQ3, we first conduct a controlled experiment to study the effect of outliers on user clicks. Next, to examine whether the findings from our controlled experiment generalize to naturalistic situations, we explore real-world click logs from an e-commerce platform. We show that, in both scenarios, users tend to click significantly more on outlier items than on non-outlier items in the same rankings. We show that this tendency holds for all positions, i.e., for any specific position, an item receives more interactions when presented as an outlier as opposed to a non-outlier item. We conclude from our analysis that the effect of outliers on clicks is a type of bias that should be addressed in ULTR. We therefore propose an outlier-aware click model that accounts for both outlier and position bias, called outlier-aware position-based model (OPBM). We estimate click propensities based on OPBM; through extensive experiments performed on both real-world e-commerce data and semi-synthetic data, we verify the effectiveness of our outlier-aware click model. Our results show the superiority of OPBM against baselines in terms of ranking performance and true relevance estimation.

4.1 Introduction

Ranking systems optimize ranking decisions to increase user satisfaction. Implicit user feedback is an important source of supervision that reflects the preferences of actual users. However, user interaction data (e.g., clicks) suffers from presentation bias, which can make its naïve use as training data highly misleading [73].

This chapter was published as F. Sarvi, A. Vardasbi, M. Aliannejadi, S. Schelter, and M. de Rijke. On the impact of outlier bias on user clicks. In *SIGIR*, pages 18–27, 2023.

Much work in ULTR focuses on position bias [3, 72, 74, 170], i.e., the phenomenon that results ranked higher are more likely to be examined and, therefore, clicked by users [72] than results ranked lower. Besides position there are several other factors that affect users' examination model and clicks [1, 33, 118, 138, 179]. In chapter 3 we have shown that inter-item dependencies can influence user judgments of relevance and the examination order of items. The existence of *outlier* items is a specific case of inter-item dependencies [138]. Outliers in a ranking are defined as items that observably deviate from the rest of the list w.r.t. item features, such that they stand out and catch users' attention. For instance, in an e-commerce search scenario, if only one item on the page features a "Best Seller" tag, it can be considered an outlier, because the tag differentiates it from the rest of the items in the ranking, thereby attracting users' attention.

Outlier bias. An outlier in a list of items can alter the examination probabilities, such that the probability of examination is higher for the outlier item (if it exists) and its neighboring items than the probability assigned by the position bias assumption [138].

Although outliers have been shown to affect examination probabilities [138], their impact on user click behavior is unknown. In this chapter, we hypothesize that clicks are biased by the existence of outliers. We refer to this phenomenon as *outlier bias* and aim to understand and address this effect by answering the following research question:

RQ3 Does outlier bias exist in click data? How can we estimate its impact and correct for this bias?

To begin, we conduct a user study where we compare the click-through rate (CTR) for specific items in two conditions: once shown as outliers and once as non-outlier items in the list. We find that users behave differently in relation to an item given its outlierness condition. The CTR of a specific item is consistently higher when it is presented as an outlier item than when it is a non-outlier item in a ranking. Next, to examine whether these findings can be generalized to naturalistic situations we perform an analysis on real-world search logs from Bol.com, a popular Europe-based e-commerce platform. The results confirm the findings of our user study. In addition, we observe that, on average, outlier items receive significantly more clicks than non-outlier items in the same lists. Moreover, users tend to interact more with lists that contain at least one outlier.

Outlier bias vs. context bias. We find that outlier bias affects user clicks such that users are more likely to interact with items that are presented as outliers, as well as their neighboring items. The closest concept to outlier bias is context bias in news-feed recommendation [179]. In the presence of context bias CTR is lower for items when surrounded by at least one very similar item than when they are surrounded by non-similar items. This is different from outlier bias, which emphasizes the *difference* between the outlier and the rest of the list. Moreover, observability is a key factor in detecting outliers in ranking as defined by [138]; this is not the case in context bias.

Accounting for outliers. Based on the findings of our user study and log analysis, we conclude that one should account for the effect of outliers when unbiasing user clicks for ULTR. To this end, we propose a click model, based on the examination hypothesis, called *outlier-aware position-based model* (OPBM), which accounts for both outlier and position bias. OPBM assumes the probability of a click depends on (i) examination,

(ii) relevance, and (iii) the outlier's position (if it exists). We use regression-based expectation maximization to estimate the click propensities based on our proposed click model, OPBM. We verify the effectiveness of our outlier-aware model for estimating propensities in the presence of both position bias and outlier bias. Following [6, 74, 116] we use a semi-synthetic setup for the experiments; the true relevance labels provided in this setup allows for evaluating the relevance estimation. Furthermore, using simulated clicks we are able to control the severity of position bias and outlier bias. The results of our experiments show the superiority of OPBM against baselines in terms of ranking performance (NDCG@10) and true relevance estimation.

Main contributions. The main contributions of this chapter are: (i) we identify and study a new type of click bias, originating from inter-item dependencies, called outlier bias; (ii) through extensive analyses of both user study results and real-world search logs, we confirm our hypothesis about the existence of outlier bias; (iii) to address this effect we propose an outlier-aware click model that accounts for outlier items (if they exist), as well as position bias; (iv) using an empirical analysis based on real-world data and semi-synthetic experiments we show the effectiveness of our outlier-aware model in estimating click propensities; and (v) we make the data from our user study plus the code that implements our baselines and OPBM publicly available.

4.2 Outliers in ranking

Outliers in ranking are items that observably stand out among the window of items that are presented to a user at once. We use the following definitions from [138] to introduce so-called outliers:

Definition 3 (Observable feature)**.** An observable item feature, $\mathcal{F}$, is a characteristic of an item in a list that can be purely presentational in nature (e.g., image, title font size, and discount tag).

Definition 4 (Degree of outlierness)**.** Let $\mathcal{M}$ be any outlier detection method, and $\mathcal{F}_i$ an observable feature corresponding to item i, in the context of all items in the list, $\mathcal{C}$. The degree of outlierness for item i is the value calculated by $\mathcal{M}$ for $\mathcal{F}_i$ w.r.t. $\mathcal{C}$ shown as $\mathcal{M}(\mathcal{F}_i|\mathcal{C})$. This value indicates how much the corresponding item differs from the other elements of the set w.r.t. $\mathcal{F}$.

Definition 5 (Outliers in ranking)**.** Let $\mathcal{M}$ be any outlier detection method; we call item i in a ranked list an outlier, if $\mathcal{M}$ identifies it as an outlier w.r.t. an observable feature, $\mathcal{F}$, based on the degree of outlierness, and in the context of the list.

In Section 4.3.2 we describe our choices of observable features and outlier detection method used in this chapter.

4.3 Impact of Item Outlierness on Clicks

In Chapter 3 we show that outlier items receive more attention from users. However, it is not known whether an item's outlierness affects users' clicks as well. To answer RQ3 we first answer the following research question:

RQ3.1 Does outlier bias exist in rankings of items?

To this end we first conduct a user study to examine the outlierness effect as the only variable factor influencing the clicks. Next, we need to examine whether the findings of our study can be generalized to naturalistic situations. In other words, we seek to establish ecological validity [9, 86]. To this end, in Section 4.3.2 we explore real-world click logs to confirm our findings.

4.3.1 User Study

In this section, we present the results of our user study. Our main goal is to learn whether the outlierness of an item affects user clicks, independent of the item's relevance and position.

Setup. We mimic Bol.com, a popular European online marketplace. We ask participants to interact with search engine result pages as they normally would, and find items they prefer and think are relevant. We focus on a list view, with 20 items on each page, and participants are able to scroll the list to see all items. We have two queries; for each query, we show one specific item once as an outlier and once as a regular item. We call this specific item the *target*, and these two variant presentations *condition I* and *condition II*, respectively. In condition I the target is an outlier w.r.t. a set of observable features, such as item category,[1] price, discount tag, and star rating. We aim to compare users' behavior between these two conditions for each query. We keep other factors such as relevance and position bias unchanged between the conditions. To eliminate the effect of position bias we always show the item at rank 4, and to maintain the same degree of relevance to the query we only change the surrounding items to change the outlierness of the target item.[2]

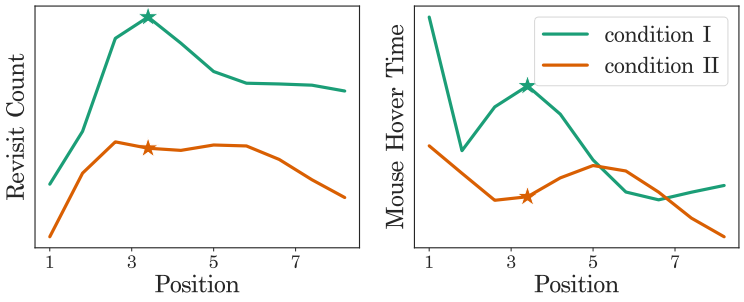
We also have a Qualtrics [128] survey. It contains the task instruction, multiple choice questions about the instructions, queries, and links to the examples, and a few demographic questions at the end. In the instructions, we describe the overall goal of the research and ask participants to read the instructions carefully. We describe what it means to interact with a result page in terms of exploring the results, scrolling the list, and clicking on items that seem interesting. Participants can click on an item to open the item's detail page. In our instructions we encourage participants to click on items they find interesting, however, clicking is not mandatory. We instruct participants to first read and understand the query, and then scan the result page as if they submitted the query themselves.

Participants. We recruit 40 workers, based in countries where our marketplace is active, from the Prolific platform [125]. From the participants, 14 are female, 23 are male, and 3 listed other genders. The majority of participants (27) are between 25 and 44 years old, with 10 participants younger and 3 older; 33 participants reported that they shop online at least once a month.

Metrics. For reporting we consider three measures based on participants' interactions with rankings: (i) revisit count, which indicates how many times on average participants

[1] Note that this feature can affect the outlierness w.r.t. the item's image as well.

[2] To examine our hypothesis about an inter-item dependency, here we assume that the relevance of a document is only dependent on the query.



(a) Revisit count (b) Mouse hovering time

Figure 4.1: Revisit count (a) and mouse hovering time (b) for the two conditions of one of our user study examples. The position of the target item is marked by an asterisk. The plots show that the user engagement with the target item is higher when presented as an outlier (condition I)

.

Table 4.1: CTR of the target item's position in both examples of our user study. The target item recieves more clicks when shown as an outlier (condition I).

	condition I	condition II
Query 1	0.944	0.166
Query 2	0.880	0.091

viewed an item (due to scrolling), (ii) mouse hover time that shows the amount of time on average participants spent on an item, and (iii) CTR for the target item in each condition, which is our main metric in this study.

Findings. We expect to see more interactions with the target in condition I. Since we keep other factors unchanged between the two conditions, we can attribute any difference in user behavior to the inter-item dependencies.

Figure 4.1 depicts the revisit counts (Figure 4.1a) and mouse hovering time (Figure 4.1b) for different positions and conditions of one example. Both plots show that the user engagement with the target item is higher when it is presented as an outlier. We see the same pattern in the second example. On average, participants revisited the outlier item more often and spent more time examining it. These findings are in line with the results of the eye-tracking experiments conducted by Sarvi et al. [138], which suggest that, on average, outlier items receive more attention from users. However, our main goal is to study whether this increased attention leads to more clicks.

Table 4.1 reports the CTR for the target item in both examples and for the two conditions. In both examples we see a large difference between the CTR reported for the different conditions, suggesting that when the target item is shown as an outlier it receives more clicks as well as more exposure.

4.3.2 Real-world Click Logs

The findings of Section 4.3.1 confirm, in a controlled experimental setup, that an item's outlierness can influence users' click behavior. However, we still need to examine the ecological validity of this hypothesis. To this end, we present our observations of click exploration of real-world search logs from our e-commerce platform. We are specifically interested in exploring the data to study the existence of outlier items in rankings and their impact on click data. Notice that we use this data only for click analysis and parameter estimation (Section 4.5).

Data collection. We collect search query logs from 20 consecutive days. Each row of the dataset consists of seven observable item features that are explained in Table 4.2, along with users' interaction signals: impressions and clicks.

Definition 6 (Impression)**.** An *impression* indicates how many times an item that is rendered by the search engine is viewed by a user. If an item is rendered in low positions, it may not end up in a window that is visible to the user, leading to zero impressions. On the other hand, the number of impressions can be greater than one due to scrolling.

We selected item features that are used across different categories, are observable by users, and have been shown by previous work to be important in influencing users' purchase decisions [4, 76]. We leave out item images from our click exploration due to the excessive complexity they would have added to this study.

Most search engines consider diversity as a quality of search result pages [5]. This can have a side effect, where the returned rankings may contain outlier items. Hence, query logs are a valuable source for studying the outliers' effect on users' clicking behavior. To begin, we define two types of rankings based on the existence of outliers as follow:

Definition 7 (Normal rankings)**.** We call rankings that contain no outlier *normal* rankings. Normal rankings can either consist of a homogeneous set of items or a diverse set.

Definition 8 (Abnormal rankings)**.** We define *abnormal* rankings to be lists that contain at least one outlier.

Outlier detection. We examine each item for outlierness based on the features described in Table 4.2 and in the context of all items in the list as described in Section 4.2. An item is an outlier if it is detected as an outlier w.r.t. at least one of these features.

We use the Interquartile rule to detect the outliers, and consider the absolute difference between the feature value and the upper/lower bound as the degree of outlierness of the corresponding item (see Section 4.2). Feature values are normalized so that we have an outlierness degree of unit range for all observable features. We set the threshold for the degree of outlierness to 0.5, which means we only label an item as an outlier if the absolute difference between its score and upper/lower bound is greater than 0.5.

Post-processing. We filter out the parts of the rankings that are not viewed by the user based on the impression signal in our data. This leaves us with the minimum ranking size of 3. However, since by definition outlierness is meaningless in lists shorter than 4, we removed these rankings from our dataset. We also removed pages with sponsored

Table 4.2: Description of the observable features used to represent the items.

Feature Name	Description	Abbreviation
price	Selling price of an item.	-
promotion tag	Universal red tag indicating various promotions, such as 'competitive price' and 'select deal'.	promotion
high discount tag	Two-piece red tag indicating high discount for an item (different from promotion).	discount
in/out-of-stock tag	Green tag indicating the in-stock or out-of-stock condition of an item.	stock quantity
users star rating	Average user star rating of the item presented by the standard 5 stars template.	rating
'select' tag	Green tag indicating that the item is a select item (similar to Amazon Prime).	select
title length	Number of tokens in the item title.	-

Table 4.3: Users' interactions with the outlier and non-outlier items, averaged over all abnormal rankings. We used Student's t-test with $p < 0.001$ for statistical significance test.

	Avg. clicks	Avg. impressions	Avg. CTR
Outliers	0.202^*	1.381^*	0.142^*
Non-outliers	0.137	1.346	0.098
Total	0.149	1.352	0.106

items to avoid any potential effect from such items on our results. The remaining 10,903 abnormal rankings have an average length of 10.24 and a median of 8.0.

Effect of outliers on CTR. In the first step of our analysis, we aim to see if users interact differently with outlier items in abnormal rankings. To this end, we look at such rankings and compare the number of interactions outliers received on average to non-outlier items in the same ranking. We focus on clicks as interactions.

Since normal rankings carry no information for our current analysis we only keep abnormal rankings. Table 4.3 shows the average clicks, impressions, and CTR of outlier and non-outlier items for abnormal rankings.[3] We calculate the CTR values (i.e., the number of clicks divided by the number of impressions of each item) per page and report the average over all rankings. Our findings suggest that both CTR and average clicks are significantly higher for outlier items when compared to non-outlier items on the same page. Moreover, we see that the number of impressions is also significantly higher for outlier items, which is in line with the finding of an eye-tracking experiment reported in [138].

Effect of outliers per position. To make sure that the higher CTR reported in Table 4.3

[3]Note that the reported values in this section are calculated based on filtered subsets of search logs, therefore, they are not representative of the true statistics of the data.

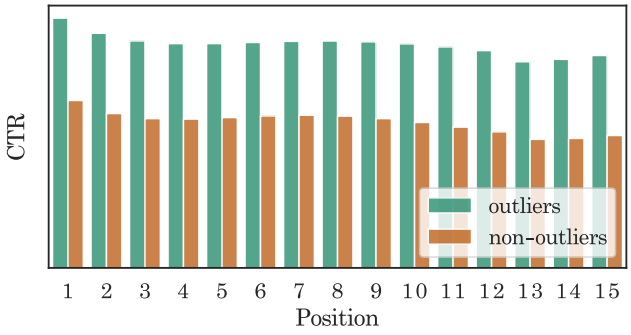


Figure 4.2: Comparison of CTR for outlier and non-outlier items per rank. CTR is consistently higher for outlier items.

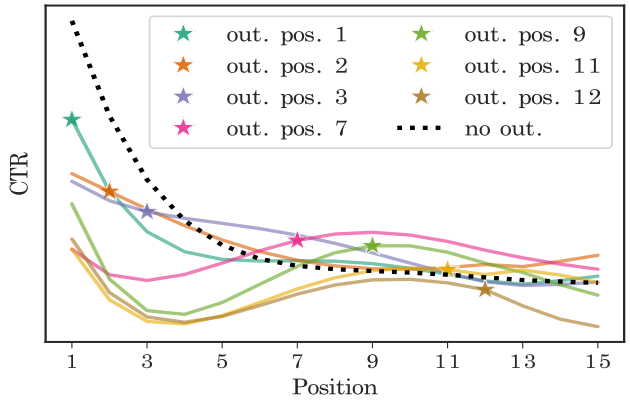


Figure 4.3: CTR per rank for abnormal rankings grouped by the outliers' position. The position of the outlier is marked with an asterisk. The values are smoothed using a Savitzky-Golay filter. Best viewed in color.

is not caused by position bias, we look at CTR values per position. Figure 4.2 depicts the results. Overall, CTR for all positions is higher for outlier items, showing that these items receive more interactions than non-outlier items.

Next, to further study how outliers change users' click behavior, we compare the CTR of the outlier position with the positions of non-outlier items throughout the ranking. To better depict the effect of outliers on different positions, we consider rankings that contain exactly one outlier; we focus on the top 15 positions. It is worth mentioning that less than 35% of the abnormal rankings in our data have more than one outlier. We group the abnormal rankings based on the position of the outlier. Figure 4.3 illustrates the results. The black line shows CTR for normal rankings. As expected this line follows position bias, where the probability of clicking an item decreases with its rank.

The other lines in Figure 4.3 show the CTR for groups of rankings with one outlier at position $r \in \{1, \ldots, 15\}$. We only show the results for some of the positions for

better visibility. We see similar patterns for other ranks. We only show the results for groups that form at least 1% of the whole collection in terms of size. In Figure 4.3 each asterisk indicates the position of the outlier. We observe that CTR distribution is different than the position-based assumption when there is an outlier in the ranking. Also, for positions after 3, we observe an increase of CTR on and around the outlier position.

Another interesting observation is that items farther away from the outlier receive less attention proportional to their ranks compared to normal pages. Moreover, we see that for positions after 3 CTR for outliers is higher than the CTR for the same position in normal pages, which is in line with our findings in Figure 4.2.

Effect of different outlier types. One can argue that different types of outliers might have different types of influence on users' perceptions. E.g., considering price as the observable feature, a very expensive outlier item might have a lower chance of being purchased compared to a cheap one. We hypothesize that these two types may neutralize each other overall in terms of statistical metrics. Hence, to examine this hypothesis, as a first attempt, we divide the outlier items into two groups of positive and negative outliers using common sense, informal definitions based on the observable features. E.g., in the previous example, the expensive item is a negative outlier while the cheap one is positive. Based on this definition, for the price feature we see that the average number of clicks for positive and negative outliers are 0.193 and 0.147, respectively; both are significantly higher than non-outlier items (0.125). We see the same trend among all observable features, both for impression and click counts. Based on these results we reject the aforementioned hypothesis and stay with our original outlier/non-outlier division.

Further remarks. We also looked at abnormal rankings in which a specific item is repeatedly shown in a fixed rank at least once as an outlier and once as a normal item. We aggregate all such rankings and observe that on average items receive 0.169 clicks in case of being an outlier, and 0.130 clicks when they are regular items in the list. Comparing the abnormal rankings to a subset of normal rankings with a similar length distribution (mean=10.09/10.30, median=8.0/8.0, std=4.95/5.82 for normal/abnormal rankings), we realize that on average the number of clicks per session is higher in the presence of outliers. More specifically, the average number of clicks is 0.139 for normal rankings, and 0.149 for abnormal rankings, with a $p < 0.001$ significance.

Upshot. To sum up, from Section 4.3.1 we learn that users behave differently w.r.t. an item given its outlierness condition (i.e., whether the item is presented as an outlier in the ranking). The CTR of a specific item is consistently higher when it is presented as an outlier item than that of a non-outlier item. Section 4.3.2 confirms the findings of our user study. In addition, we observe that, on average, outlier items receive significantly more clicks than non-outlier items on the same lists. Moreover, users tend to interact more with lists that contain at least one outlier. This section confirms the impact from outlier items on clicks. We refer to this effect as outlier bias. In the following section we propose a click model that accounts for outlier bias as well as position bias.

4.4 Outlier-aware Position-Based Model

Naïve use of implicit feedback for learning to rank can be misleading, since it suffers from presentation bias. Therefore, modeling the examination bias is crucial [46, 73].

Position-based model. Normally, items in higher ranks are more likely to be examined on a page. Position bias is formally modeled through the examination hypothesis which states that an item must be examined and perceived relevant by the user to be clicked. A widely used click model for dealing with position bias is the position-based model (PBM) [74, 170]. While being considered a simple solution, PBM is as effective as more sophisticated click models [36]. PBM assumes that the rank of an item is the only parameter that affects users' examination of that item. Examining an item means viewing and evaluating it before any subsequent interaction like a click.

Given an item d at rank k in response to a query q, the probability of clicking on d, assuming PBM, equals:

$$P(C = 1 \mid q, d, k) = P(E = 1 \mid k) \times P(R = 1 \mid d, q), \tag{4.1}$$

where $P(E = 1 \mid k)$ is the probability of user examining rank k, also called *propensity*, and $P(R = 1 \mid d, q)$ is the probability of relevance for the pair (d, q). We refer to these probabilities as θ_k and $\gamma_{q,d}$, respectively.

Outlier-aware position-based model. PBM simply assumes that the only factor influencing the propensity is the rank. In Section 4.3 we show that users are more likely to click on outlier items, hence, we assume that propensity depends also on the existence of outlier item(s).

It is noteworthy that, even among the outlier items we observe an inter-outlier position bias – the higher-ranked outlier items receive more clicks.

Hence, to model these dependencies, we propose an outlier-aware position-based model, called OPBM, that accounts for the impact of outlier items in addition to the position as follows:

$$P(C = 1 \mid q, d, k, o) = P(E = 1 \mid k, o) \times P(R = 1 \mid d, q), \tag{4.2}$$

where o indicates the position(s) of the outlier(s) in the ranking. Note that PBM is a special case of OPBM: for normal rankings OPBM is simplified to PBM.

We propose this model following Eq. (4.2) based on the assumption that the probability of examination at rank k depends on the position of outlier item(s), o, in addition to k. This model has $K \times O$ parameters, where K and O are the set of all ranks and outlier positions, respectively, which can be estimated from click data.

Propensity estimation. Here, we describe how to estimate outlier-aware position bias from regular clicks. Based on the idea of the regression-based expectation maximization (REM) algorithm [170], we propose to estimate the parameters $\theta_{k,o}$ and $\gamma_{q,d}$ simultaneously by estimating with a regression function.

Using a standard expectation maximization (EM) algorithm we aim to find the parameters that maximize the log-likelihood of the whole click logs. The log likelihood of generating click logs of the form $\mathcal{L} = (c, q, d, k, o)$ is:

$$\log P(\mathcal{L}) = \sum_{(c,q,d,k,o) \in \mathcal{L}} c \log \theta_{k,o}\gamma_{q,d} + (1 - c) \log(1 - \theta_{k,o}\gamma_{q,d}). \tag{4.3}$$

Here, we aim to estimate the parameters $\theta_{k,o}$ and $\gamma_{q,d}$ based on data points in $\mathcal{L}$. In each iteration, EM alternates between the expectation and maximization steps to compute new estimates of the parameters. In the expectation step of iteration $t+1$ we calculate the hidden variables corresponding to examination propensity (E) and true relevance (R) based on the estimated parameters at iteration t:

$$\begin{aligned}
P(E=1,R=1 \mid C=1,q,d,k,o) &= 1,\\
P(E=1,R=0 \mid C=0,q,d,k,o) &= \frac{\theta^t_{k,o}(1-\gamma^t_{q,d})}{1-\theta^t_{k,o}\gamma^t_{q,d}},\\
P(E=0,R=1 \mid C=0,q,d,k,o) &= \frac{(1-\theta^t_{k,o})\gamma^t_{q,d}}{1-\theta^t_{k,o}\gamma^t_{q,d}},\\
P(E=0,R=0 \mid C=0,q,d,k,o) &= \frac{(1-\theta^t_{k,o})(1-\gamma^t_{q,d})}{1-\theta^t_{k,o}\gamma^t_{q,d}}.
\end{aligned} \tag{4.4}$$

We then calculate the marginal probabilities $P(E = 1 \mid c,q,d,k,o)$ and $P(R = 1 \mid c,q,d,k)$ for each data point in $\mathcal{L}$. We keep the estimation of $\gamma_{q,d}$ untouched, meaning that the learning to rank (LTR) model is trained without knowledge of the outlier position and only the propensity estimation is affected by that. This leads to the maximization step at iteration $t+1$, where we update the parameters to maximize the likelihood from Eq. (4.3) as follows:

$$\begin{aligned}
\theta^{t+1}_{k,o} &= \frac{\sum_{c,q,d,k',o'} \mathbb{I}_{k'=k,o'=o}.(c+(1-c)P(E=1 \mid c,q,d,k,o))}{\sum_{c,q,d,k',o'} \mathbb{I}_{k'=k,o'=o}},\\
\gamma^{t+1}_{q,d} &= \frac{\sum_{c,q',d',k} \mathbb{I}_{q'=q,d'=d}.(c+(1-c)P(R=1 \mid c,q,d,k))}{\sum_{c,q',d',k} \mathbb{I}_{q'=q,d'=d}}.
\end{aligned} \tag{4.5}$$

The maximization step of the EM algorithm requires multiple occurrences of pair (q,d) where d is shown in different positions. To overcome the click sparsity problem and possible privacy issues, we alter the maximization step at iteration $t+1$, where we estimate the $\gamma_{q,d}$ parameter via regression [170]. Thus, given a feature vector $x_{q,d}$ representing the pair (q,d) we fit a function $f(x_{q,d})$ (e.g., gradient boosted decision tree (GBDT)) to calculate an estimate for $\gamma_{q,d}$. So, our maximization step is to find a regression function $f(x)$ that maximizes Eq. (4.3) given the estimated parameters from the expectation step. In the REM algorithm [170], this regression problem is converted to a classification problem by sampling a binary variable indicating the relevance label for $x_{q,d}$ from the distribution $P(R=1 \mid c,q,d,k)$. This results in a training set of the form $(x_{q,d}, r_{q,d})$ with the following cross entropy objective:

$$\sum_{x,r} r\log(f(x)) + (1-r)\log(1-f(x)). \tag{4.6}$$

Remark. An alternative choice instead of a single unbiased model would be to train multiple LTR models as unbiased experts for different outlier positions. This alternative has two main drawbacks. First, having experts means that each expert is trained only on

a part of the data containing outliers at a specific position. Not only can this lead to sub-optimal training, but it also makes it difficult to compare this model to the PBM-based REM as a baseline. Second, having a collection of K expert models as a ranker is not ideal in real-world scenarios. Ideally, there is a single unbiased model that can be used without information about outliers' positions.

4.5 Experimental Setup

Following much previous work in CLTR [6, 67, 74, 116, 161], we use a semi-synthetic setup for our experiments, i.e., we sample queries, documents, and relevance labels from existing LTR datasets, but simulate user clicks based on the probabilistic click models estimated on the proprietary data.

LTR datasets that contain the true relevance labels allow us to evaluate the relevance estimation of OPBM and other baselines, as well as their effect on ranking performance. Furthermore, the semi-synthetic setup enables us to control the position bias and outlier bias of the simulated clicks.

4.5.1 Data

Public LTR data. Following prior work on CLTR [74, 161, 162], we use the Yahoo! Webscope [32] and MSLR-WEB30k [127] datasets. In both datasets, there are a total of around 30k queries, each associated with a list of documents. The query-document feature vectors of the Yahoo! and MSLR datasets have dimensions 501 and 131, respectively. Both datasets have graded relevance labels with 5 levels. We follow prior work and take grades $\{3, 4\}$ as relevant and grades $\{0, 1, 2\}$ as non-relevant. The training sets of the Yahoo! and MSLR datasets have 20k and 19k queries with 473k and 2.2M documents, respectively. The test sets of the Yahoo! and MSLR datasets, have 6.7k and 6k queries with 163k and 749k documents, respectively.

Proprietary data. We use the real-world click log data as described in Section 4.3.2 for the experiments and refer to it as *proprietary* data. We use a feature vector of size 24 containing both the relevance features and products' observable features to present each query-document pair. We use these features for the LTR model. We also use the setup described in Section 4.3.2 to detect the outlier items, using the Interquartile rule, w.r.t. the observable features (see Table 4.2). Since the rankings in this dataset have an average length of 10.24 and a median of 8.0, we use the top-10 items in the experiments.

4.5.2 Click Simulation

We follow prior work [6, 74, 116, 161, 162] and sample 1% of the queries from each public dataset, uniformly at random, to train an artificial production ranker. We apply probabilistic click models on rankings produced by this production ranker to simulate clicks for the semi-synthetic experiments. We apply our outlier-aware position-based model with different approximations for examination probabilities. The relevances $\gamma_{q,d}$ are based on the relevance label recorded in the datasets. Following previous

work [74, 161] we use binary relevance:

$$P(R = 1 \mid q, d) = \gamma_{q,d} = \begin{cases} 1 & \text{if relevance\_label}(q, d) > 2 \\ 0 & \text{otherwise} \end{cases}. \tag{4.7}$$

To simulate the outlier bias we follow two strategies as follows:

OPBM$_{Real}$. We use the propensities estimated by OPBM (see Section 4.4) on our proprietary dataset. From all the abnormal rankings in our dataset, 64% contain only one outlier. Since improving ranking for more than half of queries can lead to significant improvement in real-world scenarios, we first address this majority case. Therefore, with this model, we focus on rankings with one outlier. Thus, the output of OPBM is at most a $K \times K$ matrix, corresponding to all combinations of rank and outlier position, where $K = 10$ in our experiments. We use this matrix to approximate $P(E = 1 \mid k, o)$.

OPBM$_{\mathcal{G}}$. Here, we assume that an outlier's effect on the user clicks follows a Gaussian distribution, centered at the outlier's position.

Therefore, for each k, we compute the linear interpolation of outlier bias, and position bias distributions, as follows:

$$OPBM_{\mathcal{G}}(q, d, k, o) = \gamma_{q,d}((1 - \alpha)\theta_k + \alpha\mathcal{G}(\mu = o, \sigma^2)), \tag{4.8}$$

where $\mathcal{G}$ is a Gaussian distribution with $\mu = o$, simulating the outlier effect. We set $\sigma = 1$ and experiment with varying values of α. To simulate clicks for rankings with multiple outliers, we compute the average of OPBM$_{\mathcal{G}}$ for all outlier positions (O') as follows:

$$OPBM_{\mathcal{MG}}(q, d, k, O') = \frac{1}{|O'|} \sum_{i \in O'} OPBM_{\mathcal{G}}(q, d, k, i). \tag{4.9}$$

According to our proprietary data, 91% of abnormal rankings contain at most two outliers. Therefore, in the experiments, we focus on rankings with a maximum of two outliers.

We follow previous work [67, 74, 116, 161] to define the position bias inversely proportional to the item's rank as:

$$\theta_k = \frac{1}{k}. \tag{4.10}$$

We train the LTR model[4] on 1M simulated clicks.

4.5.3 Methods Used for Comparison

Our main goal is to introduce a new type of bias and study its impact on click propensities. Hence, it suffices to compare our outlier-aware click model to baselines that only corrects for position bias. To this end, we compare OPBM with the following estimators:

- **Naïve** is a model with no correction where each click is treated as an unbiased relevance signal.
- **PBM** is the original inverse propensity scoring (IPS) estimator [74, 170] that only corrects for position bias.

[4]We use allRank implementation for our LTR [123].

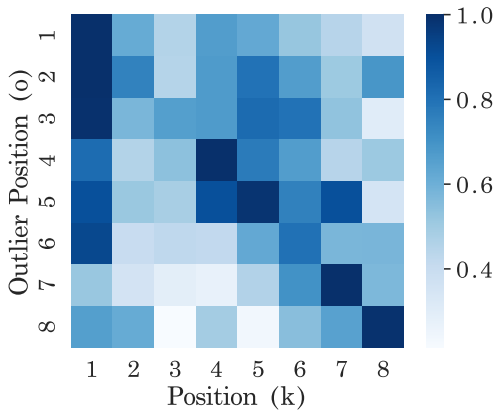


Figure 4.4: Click propensities computed by OPBM$_{Real}$ for the top 8 ranks, per outlier position on the proprietary data. Click propensities are higher on and around the outliers, contradicting the position bias assumption.

4.5.4 Evaluation Metrics

To measure the ranking performance achieved by different methods we use normalized discounted cumulative gain (NDCG). We also consider cross entropy (CE), which measures the difference between the true relevance and unbiased relevance calculated by the estimator; it is an indication of how accurately a model estimates the relevance, independent of the LTR model. Since we work with binary relevance, we compute binary CE between the corrected clicks, i.e., c/θ_k and $c/\theta_{k,o}$ for PBM and OPBM, respectively, as predictions and the true relevance values as labels. We report the mean value of CE instead of its summation, for better readability.

4.6 Results

In Section 4.3 we have already answered RQ3.1 about the existence of outlier bias in ranked lists. In this section, we continue investigating RQ3 by answering the following research questions:

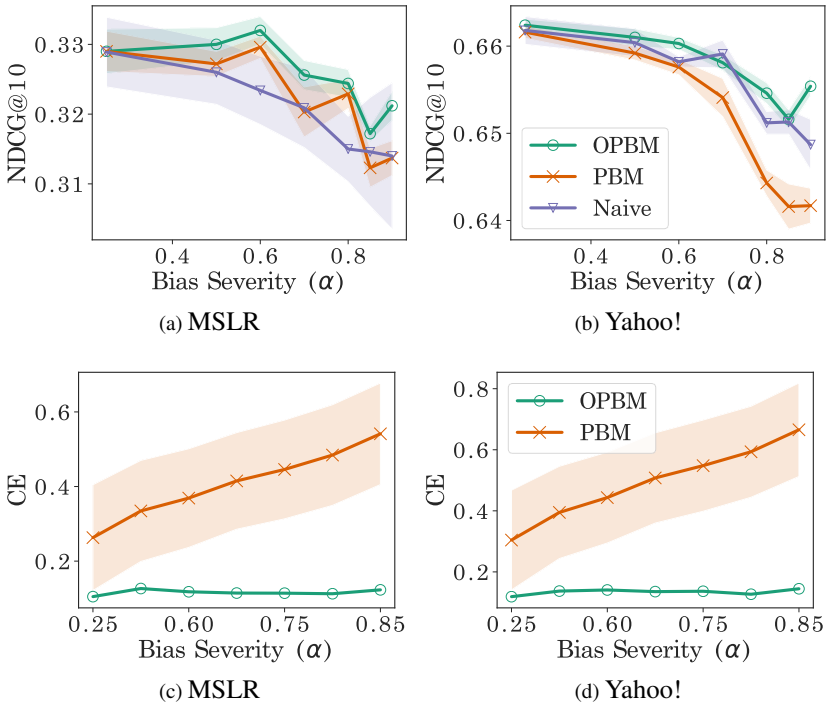
RQ3.2 How does our outlier-aware model, OPBM, perform compared to the baselines?

RQ3.3 How does OPBM perform under different outlier bias severity conditions?

RQ3.4 How does OPBM generalize to cases with multiple outliers in rankings?

4.6.1 Propensity Estimation with OPBM

We answer RQ3.2 by comparing the overall performance of OPBM in propensity estimation. Figure 4.4 depicts the propensities estimated by OPBM$_{Real}$ (see Section 4.5.2) on the top-8 ranks where a sufficient number of outliers exist in our proprietary dataset. We see that the propensities are highest on and around the outlier positions which is in



(a) MSLR (b) Yahoo!

(c) MSLR (d) Yahoo!

Figure 4.5: Comparison of different estimators in term of NDCG@10 ((a) MSLR and (b) Yahoo!) and CE ((c) MSLR and (d) Yahoo!) under varying levels of outlier bias. Results are averaged over 8 runs; shaded area indicates the standard deviation.

line with our findings in Section 4.3. However, this effect is less evident in the top-3 ranks. This is expected since we observe that position bias dominates in the top-3 ranks (see Section 4.3.2), diminishing the effect of outliers. Nevertheless, the effect of position bias decreases as the outlier appears higher in the ranking. For example, when an outlier occurs at position 1, the propensities of the first two ranks are 0.99 and 0.62, respectively. However, when the outlier occurs at position 7, these values reduce to 0.52 and 0.35.

Next, we report the results of the semi-synthetic experiments. We use the MSLR and Yahoo! public LTR datasets with simulated clicks. We use the propensities calculated by OPBM_{Real} trained on our proprietary data. We compare OPBM and PBM in terms of relevance estimation (CE) and ranking performance (NDCG@10). Table 4.4 summarizes the results; on both datasets OPBM performs significantly better than PBM in terms of CE ($p < 0.001$), indicating that OPBM approximates click propensities more effectively – it estimates true relevance of a (q, d) pair more accurately. Providing an accurate estimate of true relevance is crucial in domains such as exposure-based fair ranking [21, 107, 148], where relevance is used as an indication of an item's merit [21, 59, 107, 138, 148, 163], and can have a big impact on fairness estimation. Table 4.4 also shows that OPBM significantly improves the ranking scores (NDCG@10)

Table 4.4: Comparison of OPBM and PBM on the Yahoo! and MSLR datasets, in terms of NDCG@10 and CE. A superscript $*$ indicates a significant difference compared to the second-best performing method with $p < 0.001$.

	MSLR		Yahoo!	
	CE↓	NDCG@10↑	CE↓	NDCG@10↑
Oracle	-	0.3451	-	0.6713
Naïve	0.8205	0.3065	0.9786	0.6489
PBM	0.5474	0.3165	0.6807	0.6406
OPBM	0.1732*	0.3233*	0.1916*	0.6470*

Table 4.5: Comparison of OPBM, OPBM$_{lazy}$ and PBM on the Yahoo! and MSLR datasets, with outlier bias severity of $\alpha = 0.75$, and in terms of NDCG@10 and CE. A superscript $*$ indicates a significant difference with PBM with $p < 0.001$.

	MSLR		Yahoo!	
	CE↓	NDCG@10↑	CE↓	NDCG@10↑
Naïve	0.5704	0.3159	0.6776	0.6564
PBM	0.3126	0.3219	0.3958	0.6497
OPBM$_{lazy}$	0.1374*	0.3223	0.1548*	0.6566*
OPBM	0.1283*	0.3229	0.1407*	0.6572*

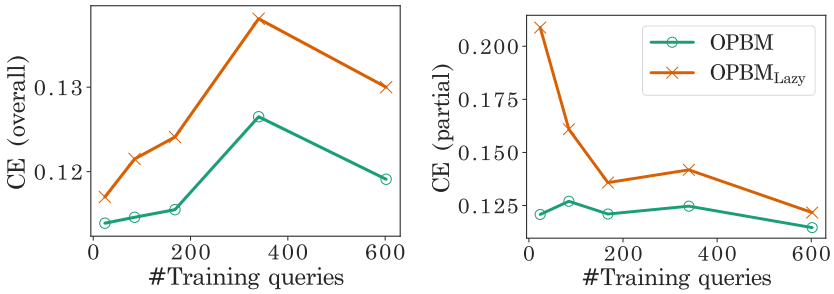
over the PBM baseline, again on both datasets.

In conclusion, using OPBM leads to more accurate propensity estimations and a more accurate approximation of the true relevance in rankings affected by outlier items. We also observe significant improvements in ranking performance by OPBM over PBM on the Yahoo! and MSLR datasets.

4.6.2 Effect of Outlier Bias Severity

Next, we address RQ3.3 by considering the impact of outlier bias severity on the performance of OPBM. For the sake of simplicity, we assume that outliers have the same effect on propensity distribution independent of their position; we use OPBM$_{\mathcal{G}}$ (see Section 4.5.2) for click simulation. The parameter α in OPBM$_{\mathcal{G}}$ allows us to control outlier bias severity. Figure 4.5 depicts the results. OPBM consistently outperforms PBM in terms of ranking performance. The results on Yahoo! dataset (Figure 4.5b) clearly show that the difference in ranking performance of the two models increases with the severity of outlier bias. In the case of MSLR (Figure 4.5a) we observe more fluctuations in OPBM's performance. This is also visible in the high variance of Naïve's performance in different runs; OPBM performs more robust compared than Naïve and PBM. Moreover, the results show that OPBM performs similarly to PBM at its worst, making it a more reliable choice as a user examination model for all severity levels of outlier bias. This is in line with our theory, which indicates that PBM is a specific case of OPBM (see Section 4.4).

In terms of cross entropy (Figure 4.5c and 4.5d), OPBM consistently outperforms



(a) Over all rankings (b) Over two outlier rankings

Figure 4.6: Comparison of OPBM and OPBM $_{lazy}$ on varying sizes of queries with multiple outliers.

PBM with a high margin. Also, the high variance in performance of PBM emphasizes the much more robust performance of OPBM compared to PBM in relevance estimation.

In conclusion, using the OPBM estimator leads to improved ranking models compared to PBM, especially when severe outlier bias exists. This finding also holds for accurately estimating the true relevance scores (i.e., CE). In the presence of slight outlier bias, OPBM exhibits a similar performance compared to PBM, making it a natural choice as it proves to be reliable.

4.6.3 Generalization to Multiple Outliers

We address RQ3.4 by considering how OPBM generalizes to multiple outliers in the ranking. For click simulation, we use Equation 4.9 with severe outlier bias ($\alpha = 0.75$). As pointed out before, our proprietary data shows that 91% of abnormal rankings contain at most two outliers. Therefore, we report results for $|O'| = 2$. Here, in addition to the single outlier rankings from the previous experiments, our semi-synthetic data contains rankings with two outliers at positions 4 and 9. As mentioned earlier, position bias is severe in the top-3 ranks, thus we place the first outlier in the fourth position of the list. Then, in order to see the effect of the outliers separately, we choose the second positions with some distance (rank 9).

To model the effect of multiple outliers, we propose two strategies: (i) According to the original description of OPBM (see Section 4.4), we consider the condition of having multiple outliers as a separate value for o, i.e., we separately compute the click propensities for k ranks, when two outliers exist in the ranking at positions 4 and 9. (ii) We simplify the problem and only consider the first outlier position, and call it OPBM $_{lazy}$. We compare the performance of OPBM between these two strategies and also with PBM. Table 4.5 summarizes the results. Overall, we see that both variations of OPBM outperform PBM in terms of NDCG@10 and CE. As expected, OPBM outperform its simpler version, OPBM $_{lazy}$ w.r.t. both metrics; we see significant improvements in CE, while the improvements over ranking performance are marginal. We can conclude that the original version of OPBM as the exact solution performs better for cases with multiple outliers. However, in case of data sparsity we can reduce the

problem to the single outlier setup and still achieve higher results than PBM.

Lastly, we provide insights into how the size of the training data influences the performance of OPBM compared to OPBM $_{lazy}$. We gradually increase the number of training queries for the rankings with two outliers (positions 4 and 9), while keeping the rest of the training set unchanged. We compare the performance in term of CE on all rankings (overall), and only on the two outlier rankings (partial). See Figure 4.6. We see that even with 24 rankings with two outliers, OPBM manages to learn the propensities better than the OPBM $_{lazy}$. However, the difference between the total performance (Figure 4.6a) of the two models grows by the size of training samples for the two outliers rankings, suggesting OPBM as a natural choice when a reasonable amount of training data is available.

In conclusion, when enough samples corresponding to multiple outlier positions are available in the training data, it is best to use OPBM with a specific o that represents the case at hand. Otherwise, reducing these samples to the single outlier setting, by only considering the first outlier position, still outperforms PBM.

4.7 Related Work

Outliers. An outlier is an exceptional object that deviates from the general data distribution [166]. Outliers can affect the statistical analysis, whether they are interesting observations or suspicious anomalies. Identifying these outlaying samples is crucial in many fields of study [89, 166]. Numerous approaches have been proposed to detect outliers [69, 89, 129, 134, 142, 191]. Defining and dealing with outliers is dependent on the application domain [166]. We follow the definition of outliers in ranking from [138]: outliers are items that stand out in the ranking w.r.t. observable item features. They study the effect of such items on the exposure distribution through eye-tracking experiments and further address the effect of outliers on exposure-based fairness. In contrast, in this chapter we focus on click bias caused by this phenomenon. We are the first to investigate the existence of outlier bias in real-world search click logs and to propose an ULTR model to correct for outlier and position bias.

Bias in implicit feedback. Users' implicit feedback, such as clicks, can be a valuable source of supervision for CLTR [2]. However, the bias in click data can cause the probability of a click to differ from the probability of relevance, which is misleading. In recent years, different types of bias have been studied, such as position [72, 74], presentation [186], selection [118], trust [2, 161], popularity [1], and recency bias [33]. Another factor influencing the perceived relevance of items is inter-item dependency [36, 138]. We introduce outlier bias, which is a type of inter-item dependency. As outlier bias considers inter-item relationships it differs from the previously mentioned types of bias. Our research suggests that users tend to interact more with outlier items such that the examination probabilities assumed by position bias change when outlier items exist in the ranking.

Presentation bias [186] considers a related phenomenon; items with bold keywords in their titles appear more attractive. This differs from outlier bias by defining attractiveness of an item independent of its surrounding items. Moreover, adding more images to the top positions in a search result page can influence CTR [104]. However, the

effect of such manipulations on click bias has not been studied. The closest concept to this research is context bias in news-feed recommendation [179]; CTR is lower for products when surrounded by at least one very similar product than when surrounded by non-similar products. This differs from outlier bias, which emphasizes the difference between the outlier and the other items. Also, observability is a key factor in detecting outliers [138], but context bias does not consider this factor. Unlike previous work, we focus on the effect of outliers on clicks, which is observable by users and comes from inter-item dependencies.

Unbiased learning to rank. Unbiased learning to rank approaches train an unbiased ranking model directly with biased user feedback [7]. These approaches can be classified into counterfactual learning to rank algorithms [6, 74, 169] and the bandit learning algorithm [115, 165, 185]. In this chapter we are concerned with CLTR. The key factor in CLTR algorithms is first estimating examination probabilities [6, 170] and then using IPS [74, 169] to debias clicks. The estimations can be derived from online result randomization [169], online interleaving [74], or intervention data harvested from multiple rankers [3]. However,interventions can hurt user experience; Ai et al. [6] propose a dual learning algorithm to automatically learn both ranking models and propensities from offline data. Similarly, Wang et al. [170] use regression-based expectation maximization to compute the likelihood of observed clicks for each query. We build on [170] and propose an unbiased ranking model that corrects for both position bias and outlier bias by adding a parameter that accounts for the position of outlier(s).

4.8 Conclusion

We have introduced and studied a new type of click bias, that is, outlier bias. We conduct a user study to compare the CTR for specific items under two conditions: once shown as outliers and once as non-outlier items in the list. We find that the CTR is consistently higher when the item is presented as an outlier than when it is a non-outlier item. Moreover, our analysis on real-world search logs confirms the findings of our user study. On average, outlier items receive significantly more clicks than non-outlier items in the same lists.

To account for this effect, we propose OPBM, a click model based on the examination hypothesis, which accounts for both outlier and position bias. We use regression-based expectation maximization to estimate the click propensities based on our proposed click model, OPBM. Our experiments show (i) the superiority of OPBM against compared models in terms of ranking performance, and (ii) that true relevance estimation outlier bias exists. We show that OPBM performs more robustly on all levels of outlier bias severity compared to PBM. Moreover, our results show that OPBM performs similarly to PBM in the worst case, making it a more reliable choice.

So far, we have shown that observable outliers in ranking affect user attention and click behavior. A natural next step is to explore how different presentational features influence the perception of outliers in e-commerce search results. This is the focus of our final chapter, where we address RQ4.

5

Understanding Visual Saliency of Outlier Items in Product Search

In two-sided marketplaces, items compete for user attention, which translates to revenue for suppliers. Item exposure, indicated by the amount of attention items receive in a ranking, can be influenced by factors like position bias. In Chapter 3 and 4 we have shown that inter-item dependencies, such as *outlier items* in a ranking, also affect item exposure. Outlier items are items that observably deviate from the other items in a ranked list w.r.t. task-specific, presentational features. Understanding outlier items is crucial for determining an item's exposure distribution.

In this chapter, we investigate the impact of different presentational features on the user's perception of outlierness to answer RQ4. By modeling the problem as visual search tasks, we compare the observability of three main features: price, star rating, and discount tag. We found that participants perceive these features differently in terms of attention and reaction times. Various factors, such as visual complexity (e.g., shape, color), discriminative item features (e.g., a solitary discount tag) and value range, affect item outlierness. These factors can be categorized into two main classes: *bottom-up* and *top-down*. Bottom-up factors are driven by visual properties such as color, contrast, and brightness, while top-down factors are influenced by cognitive processes such as expectations and prior knowledge.

In addition, we deepen our analysis of user perceptions of outliers. In particular, we focus on two key questions: (i) What is the effect of isolated bottom-up visual factors on item outlierness in product lists? (ii) How do top-down factors influence users' perception of item outlierness in a realistic online shopping scenario?

We start with bottom-up factors and employ visual saliency models to evaluate their ability to detect outlier items in product lists purely based on visual attributes. Then, to examine top-down factors, we conduct eye-tracking experiments on the same task as our visual search experiment: online shopping. This time, we design the task as a simulated e-commerce environment, mimicking a popular European online shopping platform to be more representative of real-world scenarios. Moreover, we employ eye-tracking not only to be closer to the real-world case but also to address the accuracy problem of reaction time in the visual search task. In our experiments, participants interact with

This chapter was published as F. Sarvi, M. Aliannejadi, S. Schelter, and M. de Rijke. Understanding visual saliency of outlier items in product search. *arXiv preprint arXiv:2503.23596*, 2025.

realistic product lists, some containing outliers w.r.t. different presentational features, such as image, price and discount tag, at different positions.

Our experiments show the ability of visual saliency models to detect bottom-up factors, consistently highlighting areas with strong visual contrasts and attention hotspots. While the well-known Itti & Koch model detects general visual attention patterns in an image, a graph-based visual saliency model (GBVS) identifies visual anomalies more effectively. However, one should be cautious about the limitations of these models. Visual saliency models only rely on bottom-up factors, making them naive in that they do not distinguish between separate product features or compare them against each other.

The results of our outlier-free experiment show that despite being less visually attractive, product descriptions captured attention the fastest, indicating the importance of top-down factors and user knowledge of the task. Our observations in lists with visual outliers suggest that outliers and their immediate neighbors attracted attention faster (in terms of time to first fixation), which is in line with our findings from the visual search task. However, in our eye-tracking experiments, we observed that outlier items engaged users for longer durations (in terms of fixation count and time spent) compared to non-outlier items. This effect was consistent across different outlier features (image, price, discount tag) and various positions within the list.

5.1 Introduction

In two-sided marketplaces, items compete for attention from users, since attention translates to revenue for suppliers. The exposure of the item is an indication of the level of attention that each item receives from users. Effective estimation of item exposure is crucial for challenges such as item fairness [21, 42, 103, 107, 135, 148, 149, 182] and bias in counterfactual learning to rank [2, 72, 74, 112, 118, 161]. Various modeling assumptions have been proposed for item exposure estimation in ranking. Widely used modeling assumptions made to estimate item exposure include inter-item independence and definitions of exposure as a function of an item's position in a ranking. However, recent research has introduced different types of inter-item dependencies that influence exposure distribution on a ranked list, such as attractiveness bias [186], context bias [179] and outlier bias [138, 140].

Following Chapter 3 and Chapter 4 we focus on a phenomenon that accounts for a specific type of inter-item dependency: the existence of *outlier items* in a ranked list may affect the exposure that all items in the list receive. Outlier items are those that observably deviate from the rest of the items in a ranked list w.r.t. task-specific, presentational features. Presentational features are item features that are visible to users when examining the result page, such as the price of a product in product search.

We have shown that the presence of outliers may result in attention being distributed in a different way than on a list without outliers [138, 140]. For instance, on an e-commerce search result page, adding a red-colored discount tag as a discriminative feature to only one product can attract more attention to it irrespective of its position or relevance to the query, thereby deviating from the exposure distribution that is estimated only based on position-based assumptions.

The perception and visual search communities have conducted many studies on how the human brain can immediately identify recognizable objects like outliers in an image and how different visual attributes (e.g., color, shape) can add to the complexity of this task [52, 145, 154, 175]. Since presentational features can be composed of multiple visual attributes, the relation between their use as discriminative features and their perception by users is complex.

5.1.1 Presentational Features and Attention

To gain a better understanding of the relationship between presentational features and user perception, we compare different presentational features of the e-commerce search domain. We provide insights into how different presentational e-commerce features impact users' perception of the outlierness of an item on a search result page. Informed by visual search studies, we design a set of crowdsourcing tasks where we compare the observability of different presentational features. The objective of these tasks is to find a target (i.e., outlier item) among distractors (i.e., non-outlier items), as fast as possible. Following previous work [44, 154], we use *reaction time* (RT) and accuracy in measuring the effort it takes to detect the target (outlier) among its distractors. We consider three observable features commonly used in e-commerce, viz. price, star rating, and discount tag. Previous work has shown the importance of these features in influencing users' purchase decisions [4, 76].

Our observations reveal that participants perceive different presentational features differently in terms of their attention and reaction times. In addition, we show that the visual complexity of a feature can make it more observable to users. For example, a bright red background color of a discount tag makes it easier to spot than price tags that are shown as a number with regular font size and color. These factors can be categorized into two main classes: *bottom-up* and *top-down*. Bottom-up factors are driven by visual properties such as color, contrast, and brightness, while top-down factors are influenced by cognitive processes such as expectations and prior knowledge.

In Section 5.4 of this chapter, we analyze these two types of factors separately to better understand the balance between what naturally grabs users' attention and what users prioritize during online shopping. To answer RQ4, we focus on two sub-questions:

RQ4.1 What is the effect of isolated bottom-up visual factors on item outlierness in product lists?

RQ4.2 How do top-down factors influence users' perception of item outlierness in a realistic online shopping scenario?

5.1.2 How Different Features Contribute to the Outlierness of an Item

We start with visual saliency models in Section 5.4.1 to examine the extent to which outlier products in a list can be detected based solely on bottom-up factors such as color, shape, and contrast. Next, in Section 5.4.2 we examine the effect of top-down factors by conducting eye-tracking experiments on the same task as our previous visual search experiment, i.e., online shopping. This time we design the task as a simulated

e-commerce environment mimicking a popular European online shopping platform to be more representative of real-world scenarios.

Our experiments confirm that visual saliency models are effective in detecting bottom-up factors, consistently emphasizing areas of the image with high visual contrast. While the Itti & Koch model [66] captures general patterns of visual attention in an image, the graph-based visual saliency model (GBVS) [58] is better at identifying outlier regions of the image. However, it is important to acknowledge the limitations of these models. Visual saliency models depend solely on bottom-up factors, which means they cannot distinguish between different product features or assess them in relation to each other.

Our eye-tracking experiments suggest that product descriptions captured attention quickly, although they were less visually attractive, in lists without outliers. This finding indicates the importance of top-down features and other factors in play such as center bias [28, 50, 152]. In lists with outliers, our analyses show that outliers and their immediate neighbors attracted attention faster and for longer durations compared to distant items. This effect was consistent across different outlier features (image, price, discount tag) and various positions within the list. We find that outlier items not only stand out among the list, but also receive more exposure as users spend more time examining them.

5.1.3 Main Contributions

The main contributions of this chapter are: (i) We demonstrate how different presentational features (e.g., price, star rating, discount tags) impact user perception of outlierness in e-commerce search result pages, highlighting the key role of visual complexity in attention distribution; (ii) Through experiments with visual saliency models, we analyze the influence of bottom-up visual factors on item outlierness in product lists, confirming the effectiveness of the graph-based visual saliency model in detecting visual anomalies in ranked lists; (iii) Through eye-tracking experiments, we demonstrate the impact of top-down factors on user attention, showing that these factors can override bottom-up visual signals in online shopping scenarios; (iv) We show that outlier items and their close neighbors in ranked lists attract more attention and receive increased exposure (measured by engagement time), regardless of their position, due to their distinct observable features.

5.2 Background

5.2.1 Visual Search

Visual search has been a central approach in studying visual attention for many years. It allows researchers to turn everyday search activities, like finding a can opener in the kitchen, into controlled experiments that can be repeated in a lab setting [177].

In a typical visual search task, individuals scan a visual field to locate a target object among other distracting items. Researchers examine how features such as color, shape, size, or orientation affect the speed and accuracy of finding the target [154]. One of the key theories explaining this process is the Feature Integration Theory (FIT) introduced

by Treisman and Gelade [154]. FIT proposes that visual search occurs in two stages: the pre-attentive stage and the focused attention stage. In the pre-attentive stage, basic features like color and shape are processed automatically and in parallel across the visual field. However, in the focused attention stage, when these features need to be combined to identify an object, the process becomes slower and requires more cognitive effort. This theory explains why finding a single, distinctive feature (such as a red dot among blue dots) is easier and faster than searching for an object that shares multiple features (like a red circle among red squares). Another important concept in visual search is the difference between the stand-out effect and conjunction search, introduced by Wolfe [174]. When a target differs from all distractors by just one feature (such as color), it stands out, making the search quick and independent of the number of distractors. In contrast, when the target shares features with the distractors (for example, finding a red circle among red squares and green circles), the search becomes slower, and the task duration increases as the number of distractors grows. Visual search tasks are typically assessed using RT and accuracy, which help determine how quickly and efficiently a target can be identified among distractors [99, 108, 119, 176].

5.2.2 Visual Saliency

Visual saliency determines the perceptual selection of objects or regions that stand out and capture attention within a visual field [66]. It influences the control of visual attention, for example, in determining the next fixation points during visual exploration [65, 164].

Two types of factors influence the visual saliency of an object in a specific context: bottom-up and top-down factors [82]. Bottom-up factors are primary visual attributes such as color, shape, size, and orientation. Objects that are unique with respect to such attributes tend to attain the observer's attention. For example, in an image mainly filled with green colors and shapes, the sudden appearance of a different color like red often makes people look at the red part [82, 93]. Top-down factors come from one's goals and what you expect to see [82]. They are based on one's previous experience and their knowledge of the context. For example, when searching for something specific, like a red car, one is more likely to notice red cars first.

Saliency effects can vary across different contexts and tasks. While some bottom-up factors like color contrast can be generalizable, top-down factors related to specific tasks can significantly influence what stands out. What is salient in one situation may not be in another. Researchers typically conduct specific experiments and use computational models to understand saliency within particular contexts and tasks rather than making broad generalizations [82].

5.2.3 Outliers in Ranking

Outliers are data points that differ significantly from the rest of the data [166]. They can represent unusual but important findings or potential errors. In any case, they are often seen as noise that can influence statistical analysis. Various methods have been developed to detect outliers [45, 68, 129, 142] as it is essential to detect these anomalies in many research fields [89, 166]. However, the definition of outlier items can vary

across domains [166]. In this chapter, we adopt the definition of outliers in rankings from Sarvi et al. [138] who describe outliers as items that stand out based on observable features. Observable features are visible characteristics that distinguish outlier items from their neighbors [138]. In this chapter, we create outlier items in rankings based on this definition and using different product properties as observable features.

5.3 Preliminary Experiments: Visual Search

Studies in the field of visual search and cognitive science show that different visual attributes are processed differently by the brain [153]. Inspired by these findings, and as a first step, we aim to verify if users notice deviations in different products' presentational features at different rates. In this section, we describe the details of our crowdsourcing task which is formulated as a visual search experiment.

5.3.1 Crowdsourcing Experiments

We design our tasks as a visual search process [52, 145, 154, 175], where the objective is to find a target among distractors. We focus on the domain of e-commerce search, where the distractors are non-outlier products in a ranking, and the target is the outlier products that differ in at least one presentational feature. We compare three presentational features, namely, price, star rating, and discount tag. When considering a discount tag, our task is close to a disjunctive search process known from visual search [154] that focuses on detecting a target that differs from the distractors in terms of a unique visual feature, e.g., the discount tag [101]. When regarding price and star rating, our task is closer to conjunction search [154], where the distractors exhibit at least one common feature with the target [145]. However, unlike conjunction search, in our task, the difference between the target and the distractors is in the values of the features, not the features themselves (e.g., the value of the product's price). In the rest of this chapter, we refer to the target item as *outlier*.

Following previous work [44, 154], we use *reaction time* (RT) and *accuracy* to measure the effort it takes to detect the target (outlier) among its distractors. The goal of this task is to examine and determine which presentational features are easier to detect by the workers, i.e., the shorter the RT to find the outlier, the easier it is.

We perform our experiments using two tasks, where we build several synthetic product search result pages and examine how each feature contributes to the outlierness of an item, both separately and simultaneously. Below, we describe the different stages of our two tasks.

Experimental design

In the following, we describe the details of our experimental setup.

Page examination behavior We record several signals related to participants' page examination behavior and their interaction data, such as mouse hovering, scrolling, viewed items, clicks, and time spent on the task. To gain a more accurate estimate of RT, we ask the participants to click on a *Start* button after reading the instructions. The

search result page appears only after the Start button has been clicked. We compute the task completion time from the moment the workers click the Start button.

Instructions In the instructions, we describe the overall goal of the research and the concept of an outlier in a search result page, providing tangible examples. We ask participants to scan and compare all items in a list and flag outliers as *fast* as they can. We also ask them to fill out a questionnaire after completing the tasks.

Participants We use Amazon Mechanical Turk as the platform for our crowdsourcing experiments, with workers based only in the U.S., with an approval rate of 95% or greater. After quality control, we are left with 140 assignments (92 for Task I and 48 for Task II), submitted by 80 distinct participants. From the participants, 45% are female, 53% male, and 2% listed other genders. The majority of participants (74%) are between 25 and 44 years old, with 5% younger and 21% older workers.

Post-task questionnaire We ask participants to fill out a questionnaire after completing the task. To gain more insight into workers' backgrounds and online shopping behavior, we instruct them to fill out questions on their demographics and familiarity with online shopping. Moreover, to enable more effective results analysis, we ask the workers how much each product feature influences their everyday purchase decisions. To ensure that the workers understand the outlier definition, we ask them to answer a question about the definition of an outlier in search.

Quality control We follow three strategies for quality control. As part of the post-task questionnaire, we ask a multiple-choice question on the definition of outliers. All participants managed to answer this question correctly. Also, following Kittur et al. [80], we ask workers to justify their choice in a few keywords. We only remove the responses of those participants who entered random tokens as justifications of their answers. We also remove the responses of those who revisit the instructions more than two times while performing the task, since response time is crucial in this study.

Task I

In the first task, we evaluate how fast any of the three presentational features (price, star rating, discount tag) can be spotted on a search result page. To this end, we explicitly describe and mention the one feature at a time to the participants and ask them to scan the list and find up to two outlier items, *only* with respect to the given feature. For instance, after providing a definition of outliers in the instruction, we mention that there are one or two outliers in terms of different values for price in the list and that they have to find them as fast as they can. We place one of the outliers at the top of the list and the other at the bottom. To avoid position and randomness bias, we keep the position of the outlier items fixed while other items are randomly placed in the list.

Wolfe [175] suggests that visual features including luminance, color, and orientation affect the RT in a visual search task. Following this work and inspired by the experiments in [154], we tested two variations of Task I, namely Type I and Type II, where we change the shape, color and value of the presentational features to study different magnitudes

of deviation of the outliers from the rest. In Type I, we use features that more strongly discriminate between the outlier and the rest compared to Type II. For example, an outlier w.r.t. price can be 10 or 2 times more expensive than other products. We use the former in Type I and the latter in Type II. The same goes for star rating. Regarding the discount tag feature, we use the suggestions by Wolfe [175] to distinguish between the outlier items of Type I and Type II. In Type I we use a bright red color as background with a bold white font stating that there is a special deal on the product, whereas, in Type II, we use a light green text without any background stating a 10% discount.

Task II

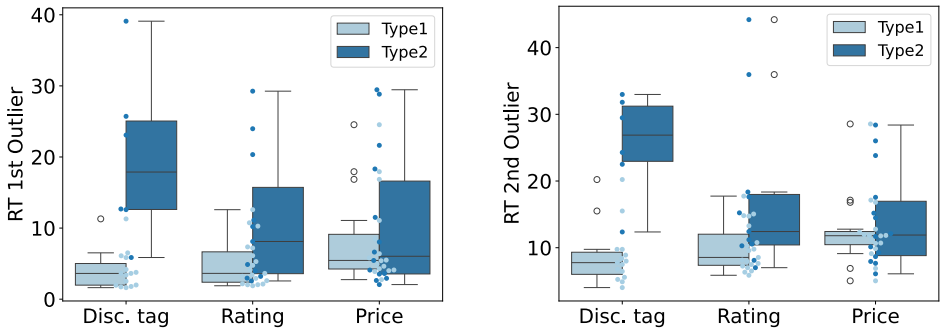
Unlike Task I, here we aim to examine the relative RT for the three features (price, star rating, discount tag) when presented to the users simultaneously. To better compare the three observable features, we jointly present the different combinations of these features and analyze the behavior of the users. While describing the three target features in this task, we do not mention to participants which features are being examined. Therefore, the workers are supposed to go through the list, examine all items with respect to all the features used in presenting the results, and then decide which items are outliers. Note that there are more than three features used to describe each item, for example, we used image, title and delivery information next to the price, discount tag and star rating. Moreover, we indicate that the workers have to mark a maximum of three items as outliers. Here, we also randomize the position of the outlier items while making sure that they appear both at the top and bottom of the list. We run the task for all combinations of at least two of the three features.

5.3.2 Results

In this section, we present the results of our crowdsourcing tasks in terms of the performance and behavior of the workers under different experimental conditions.

Table 5.1: Worker performance metrics in terms of RT and accuracy. Average (Avg.) and median (Med.) RT in seconds is reported for the first and second outlier (out. 1 & out. 2).

Type		**RT out. 1**		**RT out. 2**		**Acc.**
		Avg.	Med.	Avg.	Med.	
Type 1	Disc. tag	4.22	3.62	8.90	8.00	0.98
	Star rating	4.81	3.41	9.67	8.14	0.97
	Price	8.38	5.50	12.44	11.77	1.00
Type 2	Disc. tag	19.84	17.88	25.57	26.88	0.99
	Star rating	10.96	8.11	17.36	12.42	0.99
	Price	10.62	6.03	14.21	11.90	0.98



(a) (b)

Figure 5.1: Distribution of the RT for the (a) first and (b) second outliers in both variations of Task I.

Reaction time and accuracy

Following [44, 154], we use reaction time (RT) and accuracy to measure the effort it takes to detect the outlier among non-outlier distractors. Table 5.1 summarizes participants' average responses to Task I in terms of RT and accuracy. Accuracy is high for all variations of Task I with a maximum of 1.00 for price in Type I and a minimum of 0.97 for star rating. We conclude from the high accuracy values that the workers grasped the concept of outlier in a ranked list and were able to accurately find them in the list. Next, we compare the time that the workers take to spot the outliers. Table 5.1 shows that for Type I outliers participants spotted the discount tags faster than the other two features in Task I. This is followed by star rating with a slightly higher recorded RT. As expected, we see that on average it took participants almost twice as long to find the price outliers. We conducted a one-way ANOVA test on RT for first outliers in Type I. Results show that the differences are statistically significant with p-value < 0.05.

Detecting discount tags is similar to a disjunctive visual search process, which has been shown to be easier to solve compared to conjunctive search (i.e., star rating and price) [154]. Moreover, users favor simple options when they act under time pressure [15], which can lead to being biased towards easy-to-detect visual features such as discount tag. The higher RT for price can be attributed to the fact that certain visual features, including color and shape, are processed early in the brain using pre-attentive processes [154]. Star rating and discount tag have more visual characteristics regarding shapes and colors, however price is more simply presented in the product descriptions.

Another related aspect is the unknown range of the price values. This is less crucial for star rating or discount tag since the former has a range between 1 to 5 and latter is a binary feature.

Type I vs. Type II

Next, we compare the results of Type I and Type II outliers. Our goal is to understand how much changing the magnitude of the deviations in terms of different features affect

user performance. One can compare different ranges of deviations on specific features to model the relationship between the the deviations and user performance, but we leave this as future work and only compare two variations. The results in the upper and lower parts of Table 5.1 suggest that the reduced magnitude of deviations in all features leads to higher RT. Duncan and Humphreys [44] study the same effect by pointing out that when outlier to non-outlier similarity increases, the task becomes more difficult. Similarly, we see that RT increases for all the features, and for both the first and second outliers.

Moreover, we see in Figure 5.1 how the RT distribution of the two outlier variants differ for Task I. As expected, the plots show a higher RT for all features, and both outliers. However, it is interesting to note that we observe the lowest relative effect on the price (26.73% increase), compared to star rating (127.86%) and discount tag (370.14%). We relate this to the visual nature of discount tag and star ratings. Reducing the color contrast of discount tag would have a greater impact on the user's ability to detect it among the distractors, compared to a different price ranges, as the user still has to carefully check the prices to detect the outlier. Regarding the accuracy, we see no drop, suggesting that even a more subtle deviation in observable feature can be detected by many users.

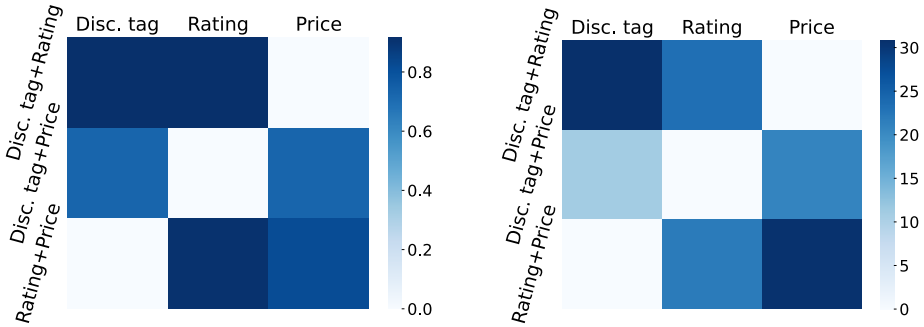
Feature combinations

Figure 5.2a shows the recall values for combinations of features, where the y-axis indicates recall of a combination of features and the x-axis indicates the value for a specific feature. As expected, detecting the outlier w.r.t. price is more difficult for participants (on average, 1.3% and 7.3% lower values than for the discount tag and star rating). In terms of RT, Figure 5.2b confirms our findings in Table 5.1, except for the combination of discount tag and star rating, where on average participants found star rating ~ 7.5 seconds faster than discount tag. Perhaps, it is because the average position of the outlier w.r.t. discount tag is lower than the star rating (12 and 9.5, respectively).

5.4 Extended Experiments: Visual Saliency and User Attention

In the previous section, we have investigated how individuals perceive and react to different product features (whether outliers or not) within a list through controlled visual search tasks. We measured participants' RT and accuracy as they detect outliers among regular items in ranked lists. The findings revealed that various factors, such as visual complexity (e.g., shape, color), discriminative item features (e.g., a solitary discount tag), and value range, affect user perception of item outlierness. These factors can be categorized in two main classes: bottom-up and top-down. To better understand the balance between what naturally grabs users' attention and what users prioritize during online shopping, in this extended study, we analyze these two types of factors separately by answering RQ4.1 and RQ4.2.

We start with visual saliency models in Section 5.4.1 and examine the extent to which outlier products in a list can be detected based solely on bottom-up factors such



(a) (b)

Figure 5.2: (a) Recall and (b) RT for combinations of observable features. Y-axis shows the metric of the corresponding feature and x-axis shows the second feature used in the combination.

as color, shape, and contrast. Next, in Section 5.4.2 we examine the effect of top-down factors by conducting eye-tracking experiments on the same task as our previous visual search experiment, i.e., online shopping. This time we design the task as a simulated e-commerce environment mimicking a popular European online shopping platform to be more representative of real-world scenarios. Formulating the task as a visual search experiment is valuable for gaining initial insights into user behavior and attention patterns, however, it has two main limitations that we aim to address in our new setup.

Representativeness of e-commerce scenarios Visual search tasks are structured and goal-driven, where participants are explicitly instructed to fulfill a specific goal (in our case: find the outlier item as quickly as possible). However, in real-world online shopping, users engage in more open-ended exploration, distributing their attention across multiple items and features without a clear target in mind. This contrast in task characteristics can lead to variations in user behavior, making it challenging to generalize findings from controlled visual search experiments to the real-world shopping case. In this section, we employ a more realistic experimental design that mimics a popular e-commerce platform in Europe.

Accuracy of RT Much previous work uses RT as a proxy to estimate user attention in visual search tasks [99, 108, 119, 176], however, it might not always be accurate because it relies on participants' interpretations of our instruction. While we instructed participants to find the outlier as fast as they could, it is not straightforward to ensure they consistently follow this instruction. Response times can be influenced by factors beyond attention, or individual differences, such as external environmental distractors or lack of focus and motivation. Therefore, RT alone may not provide a precise measure of attention.

In this extended study, we use eye-tracking as the tool for capturing and analyzing user attention. Attention, by its nature, is an internal and subjective experience [35,

144, 147]. Declarative methods, like surveys or self-reports, are inadequate when used alone for measuring user attention [13]. Eye-tracking provides objective and direct measurements of attention, making it a more reliable choice for understanding how users engage with visual stimuli [13, 35].

5.4.1 Visual Saliency Maps

Visual saliency determines the regions that stand out and capture attention within a visual field [66]. In our task of detecting outlier product feature in search result list, it allows us to predict which product features are expected to attract more attention due to their visual properties, such as color contrast, size, or pattern complexity. Visual saliency can serve as a benchmark or baseline for comparing the inherent attention-grabbing properties of different product features. It can also detect observable outliers purely based on bottom-up factors. In this section we aim to answer RQ4.1. In the rest of this section we describe the models used for this experiment and our stimuli design.

Models Visual saliency models create density maps that depict the extent to which individual pixels grab attention relative to others (see Figure 5.9). These models can be classified into bottom-up and data-driven. Bottom-up models are based on primary visual attributes such as shape and color, while data-driven models are trained using eye movement data together with some architectural assumptions inspired by bottom-up models [82]. To answer RQ4.1 we use two well-known bottom-up models that are often used as baseline models in the literature: Itti & Koch [66] model and graph-based visual saliency (GBVS) [58] model. Itti et al. [66] proposed a saliency-based visual attention model to extract visual features as computed via linear center-surrounded operations with Gaussian pyramids for intensity, color, and orientation. The GBVS model is an extension of the Itti & Koch model that forms graph-based activation maps from visual features and normalizes them to highlight conspicuity. The global visual feature extraction and graph-based activation maps enable the model to capture saliency maps at the global level.

Stimuli description To answer RQ4.1 we captured high-resolution screenshots of search result pages of Bol.com, a popular European e-commerce platform. Each screenshot is preprocessed to fit the input requirements of the saliency models. We then generated saliency maps using both models to highlight areas predicted to attract the most visual attention.

For the analysis, we selected a subset of variants of product lists that include different product categories, different outlier positions and outlier types:

1. a list of smartphones with an outlier image at position 3 (see Figure 5.9a),
2. a list of monitors with an outlier discount tag at position 8 (see Figure 5.9d), and
3. a list of office chairs with an outlier price at position 13 (see Figure 5.9g).

We generate the visual saliency maps for all these variants, illustrating both the full list (see Figure 5.9) and a focused view around the outlier and its close neighboring items (see Appendix, Figure 5.10).

5.4.2 Eye-tracking Experiments

In this section, we describe our experimental design to answer RQ4.2 through eye-tracking. We detail our methodology, experimental designs, and specific study goals.

Experimental design

In the following, we outline the key aspects of our experimental design.

Online shopping experience There are two critical factors influencing a customer's shopping experience: their goal or specific task, and product category [35].

Customer's goals refer to the different stages of the purchasing process, such as gathering information about products, comparing different options, and understanding delivery choices [157]. In this study we focus on the *Choice Task* as described by Chocarro Eguaras et al. [35]: "Visit the website and select from those offered the product that most appeals to you based on the information provided."

In addition, the category of the product has been recognized as a significant moderator in e-commerce. Nelson [110] divides product categories into two classes: search products and experience products. *Search products* are items that consumers can determine most of their attributes before purchasing. On the other hand, most attributes of *experience products* are unknown to consumers before the purchase or the consumption. In short, consumers can evaluate *search* products by their features, brand, or price, while *experience* items need senses for their evaluation [34, 110, 171].

Previous studies suggest that user attention patterns can be different for different product categories [81, 94, 168]. Therefore, following previous work [35, 62, 79, 83, 94], we select 5 product categories with search and experience attributes:

- *experience* attribute: backpacks, office chairs, running shoes; and
- *search* attribute: mobile phones, monitors.

We use the backpacks category also for the calibration stage and the rest for actual analysis.

Stimuli description The stimuli in this study consist of product search result pages, mimicked after Bol.com (see Figure 5.9). Each page contains 15 distinct products. These products are characterized by various features, including product images, titles, descriptions, star ratings, review counts, pricing information, and, in some cases, discount tags. To reduce the complexity of the page we removed information about delivery, seller, and offer from the product features.

The products on our simulated search result pages are real items from Bol.com, chosen to match our study queries. However, they may not be exactly the same as the platform's search results. We made controlled modifications to certain product features for the purpose of our research. For instance, we may have adjusted item prices or added discount tags to items to study their effects. We provide the descriptions of these decisions where they were applied.

Online eye-tracking We use the `RealEye.io` online platform to run webcam-based eye-tracking experiments. `RealEye.io` eye-tracking is based on WebGazer, an eye tracking JavaScript library [122]. Webcam-based eye-tracking has become popular in the eye-tracking community due to its ability to capture eye movements in real-world settings, relatively low cost, and high speed of data acquisition [172]. Several recent studies use webcam-based eye-tracking for their perception and cognitive experiments [26, 47, 48, 57, 109, 138]. Wisiecka et al. [172] report that they were able to obtain comparable results from `RealEye.io` compared to a lab experiment in tasks involving fixation (location-based metrics). The average accuracy for individuals is reported as 113px; however, it is expected that the average error goes to zero in aggregated analysis with several participants.[1]

Metrics and data collection In this study, area of interest (AOI) is an analytical tool that provides eye movement metrics for user-defined areas of an image. We defined three AOIs per product in the list that covers a product's image (on the left), the product's description (in the middle) and the product's price information (on the right) including discount tags if any. We consider four eye-tracking measures to report our results based on participants' eye fixations and for any AOI [8, 49]:

1. fixation count: the number of fixations within an AOI; more fixation means more visual attention;
2. time spent: shows the amount of time that participants spent on average looking at an AOI;
3. time to first fixation (TTFF): the amount of time that it takes participants on average to look at the AOI for the first time; and
4. revisit count: indicates how many times participants looked back at the AOI on average.

To calculate these metrics, we aggregated eye movements on each AOI on the list.

Instructions Participants were instructed to interact with the presented product lists as they typically would. While participants were not obligated to click on any items, they were encouraged to explore the entire page thoroughly. It was emphasized that we will ask questions regarding the content of each page after their exploration and we only accept submissions with reasonable answers to the questionnaires. We had 6 product lists each corresponding to a unique query, and participants had 90 seconds for exploring each single page. The first list was for calibration, so that the participants get familiar with the format of the experiment, therefore, we recorded the results only for the next 5 lists.

Post-task questionnaire Our questionnaire consisted of two sets. The first set was presented after each page was shown, focusing on the page's content, the products participants observed, their purchase recommendations, and whether they noticed anything

[1] `https://support.realeye.io/realeye-accuracy`

unusual or intriguing. The second set of questions was presented upon completion of the task and covered participants' demographics and online shopping habits.

Procedure and implementation Participants were recruited through the Prolific platform[2] and, after receiving task instructions, were directed to RealEye for the eye-tracking experiments and questionnaires.

The participants were randomly exposed to different sets of lists. Each set contains the same queries and products, but with modifications to place outliers in different positions. We also applied randomization within the set, to present different orders of queries to different participants. The randomization of each set presented to participants was systematically managed through our backend logic, while the randomization of the order of lists within the sets is handled by the RealEye platform.

At the start of the task, each participant would initiate their session with a warm-up step, during which they explored a list of products. The results of this warm-up step are excluded from our subsequent analysis. Upon completion of all sessions and questionnaires, participants received a unique code, allowing Prolific to track their submissions.

Participants and procedure To ensure the familiarity of participants with the e-commerce platform, we only hired participants from the Netherlands and Belgium, where our e-commerce platform is active. To ensure data quality, we require that workers have an approval rate of 95% or higher. After quality control procedures, we are left with a total of 118 distinct participants. 54.2% of participants identified as female, 45% as male, and 0.8% selected other genders. In terms of age distribution, the majority of the participants (67.8%) were 18 to 34, 30% were 35 to 54 years old, and the remainder were older than 54. All participants in our study used desktop computers with a webcam, ensuring a standardized viewing experience between all participants.

Task I

Overview In the first step of our eye-tracking experiment, we investigate how users examine product search result lists and understand the attention dynamics related to different product features. Specifically, we explore which features are more engaging in terms of time spent and fixation count and which ones capture attention faster in terms of TTFF. While there are studies examining factors that influence users' viewing behavior on search results pages [84], to the best of our knowledge, no previous research has focused on e-commerce.

Stimuli description As mentioned in Section 5.4.2, the product lists used in our study were harvested directly from Bol.com. Each list contained 15 distinct products. To maintain consistency for this step, we ensured that no outlier products were present in the lists. To this end, we identified and replaced products that could potentially be perceived as outliers (see Section 5.2.3). Several factors were considered during this process, including product images (content, color, background color), prices, discount

[2] https://prolific.io

tags, and user ratings. Additionally, slight adjustments to item prices are applied in some cases to ensure they match the pricing patterns found in product lists. We used "backpacks" and "running shoes" as the search queries for this experiment. Participants were randomly assigned to one of these lists.

Task II

Overview In our second eye-tracking study, we aim to investigate how outlier product features influence the observability of products in search result lists. More specifically, we focus on the stand-out effect of different product features when presented as outliers. We aim to answer RQ4.2 by exploring how the presence of an outlier product feature affects the overall attention distribution among the list and how these effects differ among various product features.

Stimuli description We selected three product features for examination: price, image, and discount tag.

To maintain an unbiased experimental design, we ensured that product category, position, and relevance do not introduce any bias into the results. To achieve this, we created unique combinations of product queries and product features, each featuring an outlier placed at positions 3, 8, or 13 within the product list.

Each participant viewed each product query only once during the study, ensuring diversity and preventing familiarity from affecting their attention patterns. Furthermore, participants received lists with outliers w.r.t. each product feature only once to prevent repetition bias and learning effects. The position of the outlier within the list was randomly assigned from the available options to address position bias.

5.4.3 Results

Visual saliency maps

In this section, we present our observations to answer RQ4.1 using the full list view, however, the patterns are similar for the focused view images (see Figure 5.10). For the first list, the GBVS map highlights a very intense spot near the top where the outlier image is located (see Figure 5.9c). This suggests that the model is highly responsive to the visual characteristics of the outlier, potentially due to its unique color scheme, size, and contrast compared to surrounding items. On the other hand, the Itti & Koch map shows a more evenly distributed pattern of saliency across the product list (see Figure 5.9b), with less intense focus on any single point. However, there is still a noticeable emphasis on the area around the image outlier, but less pronounced than in the GBVS model.

In the second list, the GBVS map shows several highlighted areas, but there is a particularly intense focus on the upper part of the list, where the initial items are located (see Figure 5.9f). The model does not distinctly highlight the middle part where the outlier discount tag is at position 8, suggesting that while the GBVS model is sensitive to certain visual cues, it may not consistently emphasize elements like tags unless they are accompanied by other strong visual contrasts. While the Itti & Koch map displays attention points scattered more evenly across the entire list, there are visible

highlights around the middle section, closer to where the outlier discount tag is (see Figure 5.9e). The highlight around the outlier discount tag in the middle of the page might not specifically show the outlier; however, we observe that the Itti & Koch map could better detect this local difference in the map. This can be due to its algorithmic sensitivity to a broader range of visual features beyond mere contrast, such as layout structure.

Lastly, in the list with the price outlier, the GBVS map shows a few distinct areas of high saliency, with notable intensity at the bottom of the list, near where the outlier is located (see Figure 5.9i). This suggests that the GBVS model is effective in identifying significant deviations in price among this list, while the Itti & Koch model fails to detect this pattern (see Figure 5.9h).

In conclusion, our observations suggest that the GBVS model is particularly suited for detecting distinct visual anomalies within a cluttered visual field, while the Itti & Koch model offers insights into general visual attention patterns across a product list, by highlighting the more visible parts of the image without focusing on global differences. The GBVS model focuses on global visual features, which means it responds differently based on the complexity and variety of elements in an image. This behavior helps explain why the GBVS model fails to detect the bright red tag in Figure 5.9d, while it effectively highlights the price outlier in Figure 5.9g. In the chair list, the uniform dark colors of the product images make even small variations in price patterns more noticeable to the model. Conversely, the diverse colors among the monitor images might distract the model, causing it to overlook the red tag despite its visual prominence.

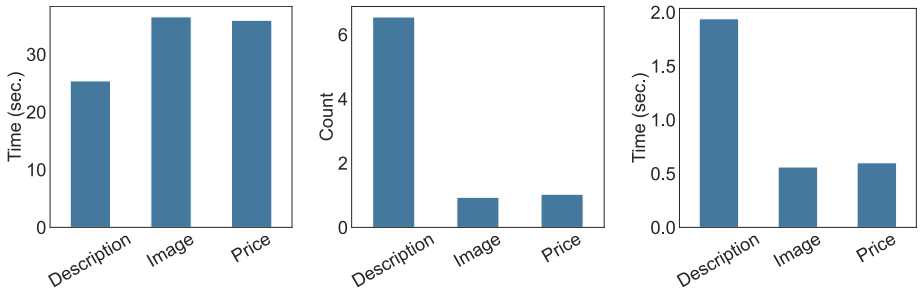
While the Itti & Koch and GBVS models provide valuable insights into the potential attention-grabbing properties of various product features, they are based on theoretical constructs and may not fully capture real-world user behavior. To address this gap, we now turn to empirical validation through eye-tracking experiments.

Eye-tracking Task I

In this section we present our observations on how users examine a regular product result page without any outliers.

Engagement metrics Figure 5.3a shows that TTFF is quite high across all categories, with product description being the fastest (25,393 ms), followed by prices (35,901 ms) and images (36,503 ms). This might indicate that the participants take a significant amount of time before they fixate on any specific element, starting from the middle of the page, confirming center bias [28, 50, 152]. Based on a Kruskal-Wallis test [102] there are statistically significant differences in the TTFF across the three feature categories (p-value < 0.05); however, the difference between TTFF on product description and the two other features is more noticeable which could be due to the complexity of scanning the textual information.[3] The average fixation count (see Figure 5.3b) is also significantly higher (Kruskal-Wallis test, p-value < 0.05) for product description (6.55 times) compared to price (1.03 times) and image (0.94 times), suggesting that once users engage with detailed text, they tend to revisit or focus on these areas more frequently, potentially reflecting a deeper cognitive processing or evaluation.

[3] Keep in mind that we instructed users to carefully examine the products.



(a) (b) (c)

Figure 5.3: (a) TTFF , (b) average fixation count , and (c) average time spent by product feature category.

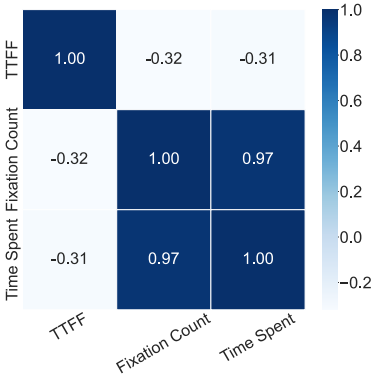


Figure 5.4: Correlation matrix of user engagement metrics based on the eye-tracking experiment.

Consistent with the fixation count, users spend significantly more time (Kruskal-Wallis test, p-value < 0.05) on product description (1,939 ms) than on price (600 ms) or image (561 ms) (see Figure 5.3c). This indicates that detailed textual information holds user attention longer.

Moreover, we conducted a correlation analysis between different eye-tracking metrics to see if items that capture attention faster also tend to engage users for longer. Figure 5.4 shows the results. The fixation count and total time spent on different AOIs demonstrated a strong correlation (r = 0.80), suggesting that areas that attract more fixations typically engage users for longer duration. This relationship highlights the engagement potential of product description over image or price.

Positional impact analysis To understand how the position of an item within a product list affects user interaction, we analyzed TTFF, fixation count, and total time spent based on the item's position in the list. Figure 5.5a shows that TTFF tends to increase with the position of the item in the list as expected, indicating that users examine the list from top to bottom, and items later in the list take longer to attract the initial fixation.

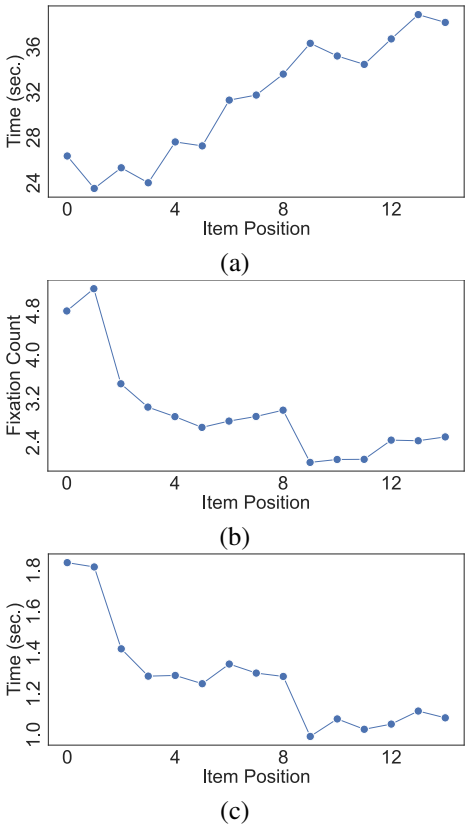


Figure 5.5: (a) average TTFF, (b) average fixation count and (c) average time spent per position.

The number of fixations generally decreases from the start towards the middle of the list and slightly increases towards the end (see Figure 5.5b). This pattern could be influenced by how users scan the page, possibly scanning more quickly through middle items after initially examining the first few items more thoroughly.

Similar to fixation count, the total time spent also decreases through the list, with the least amount of time spent around the middle of the list (see Figure 5.5c). This suggests less engagement with items as users move through the list.

In general, these trends indicate that position affects how users interact with items in a product list. Items placed at the beginning of the list are likely to capture attention faster and engage users more than those placed in the middle or bottom of the list.

Eye-tracking Task II

To answer RQ4.2 we investigate how outlier product features impact the visibility of products in search result lists. This section reports on the findings related to how different features, when presented as outliers, affect visual attention across different list positions. We answer the following questions to address RQ4.2:

RQ4.2.1 How do outlier features in product lists influence the initial user attention?

RQ4.2.2 What is the impact of outlier features on user engagement?

RQ4.2.1 We examined how the presence of outlier features in product lists affects initial user attention. Specifically, we focus on TTFF to understand whether products with outlier features attract attention faster.

In previous chapters, we have shown that both outliers and their close neighbors attract user attention faster [138]; therefore, to answer RQ4.2.1, we compare TTFF for products with outlier features and their immediate neighbors against more distant neighbors. Immediate neighbors are defined as the products immediately preceding and following the outlier, while distant neighbors are those either before or after the immediate neighbors. We analyze this setup across the outlier features (image, price, discount tag) and different product positions (3, 8, 13).

For image outliers, Figure 5.6a shows that outliers and their immediate neighbors at all positions generally exhibit a lower mean TTFF compared to distant neighbors, indicating quicker attention capture ($25.76s$ vs. $31.24s$ for position 3, $31.77s$ vs. $43.71s$ for position 8 and $34.57s$ vs. $37.03s$ for position 13). Similar trends are observed with price and discount tag features (see Figure 5.6b and Figure 5.6c), where outliers and their immediate neighbors consistently show lower TTFF compared to distant neighbors, although differences were less pronounced compared to image features. A Kruskal-Wallis test shows statistically significant differences between TTFF of the two groups among different outlier features and positions with p-value < 0.05. Figure 5.7 details the distribution and range of TTFF values.

Our results confirm that outlier features not only draw attention more quickly but also potentially increase the exposure of their adjacent items. This effect is consistent across different types of features and various positions within the list.

RQ4.2.2 To address RQ4.2.2, we analyze how outlier features influence user engagement by examining metrics such as total fixation count, time spent, and revisit counts. These metrics help us understand not just the initial attention (as explored in RQ4.2.1) but also the sustained interest and engagement. Figure 5.8 depicts the average values of these metrics calculated over all users for different features, outlier positions, their immediate and more distant neighbors.

Starting with the total fixation count, we observe peaks at positions 3, 8, and 13 for image outliers, indicating significant engagement at all positions, but particularly strong at position 8 where the average fixation count reaches ≈ 2.5. This suggests that image outliers, regardless of their position, tend to draw consistent attention, with a focus on the middle of the list. Price outliers show similar peaks at these positions with the highest at position 3 (≈ 2.75), indicating that price outliers at the start of the list may capture slightly more attention than those later, possibly due to immediate price evaluation when beginning the list browsing. Discount tag outliers exhibit the highest fixation count at the start of the list (position 3 with a count of about 4), with noticeable decreases thereafter. This highlights that discount tags catch the eye quickly, possibly due to the initial scanning behavior of users.

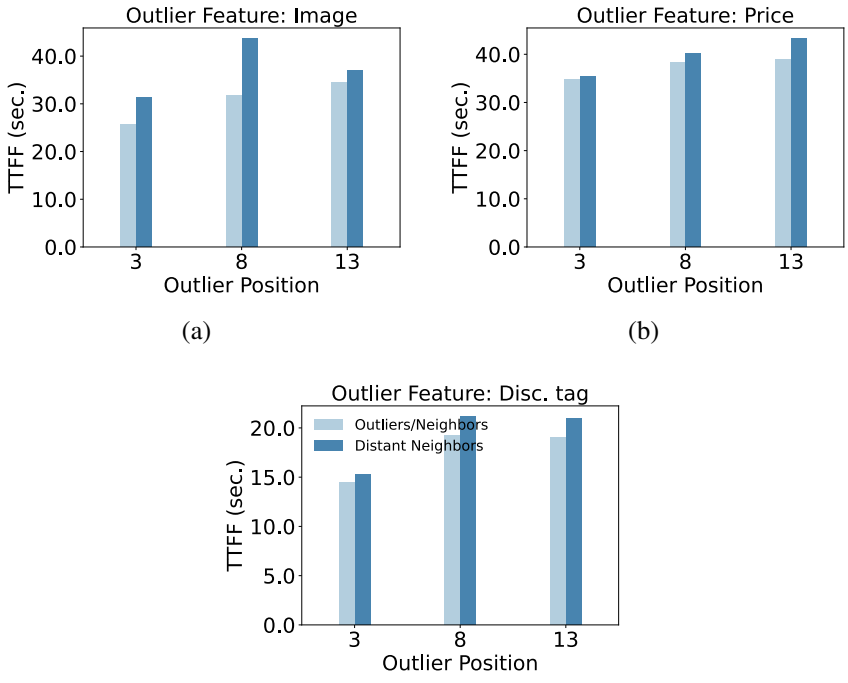


Figure 5.6: TTFF for (a) image, (b) price, and (c) discount tag outliers per position, showing the comparison between outliers and their immediate neighbors versus their distant neighbors.

Time spent and revisit counts, follow the same trends for different variants, emphasizing that outlier items and their immediate neighbors not only capture user attention faster than other items in the list, but also receive more exposure and user engagement.

Comparison and upshot

The Itti & Koch and GBVS models predict attention hotspots using visual features like color, contrast, and size. The Itti & Koch model provides a more evenly distributed attention pattern, recognizing areas with notable visual differences without intense focus on a single point. In contrast, the GBVS model highlights prominent anomalies with strong visual contrasts, successfully identifying image, and price outliers due to their unique visual characteristics.

However, these models fail to consistently detect outliers in the product lists. This limitation can be attributed to the fact that the models only rely on the bottom-up factors, which has two main implications. First, the models see the whole list as one picture. For instance, in our experiments, the diverse colors in the monitor list distracted the model from the red tag (see Figure 5.9f), while the uniform dark colors in the chair list made small price variations more noticeable.

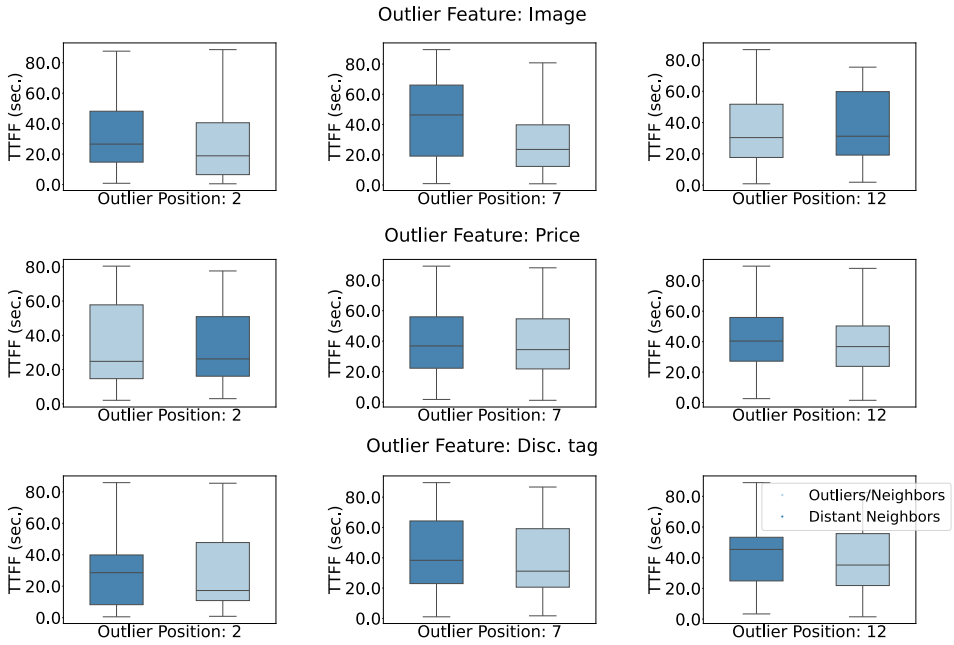


Figure 5.7: TTFF distribution across positions and outlier features.

Second, the models ignore top-down factors such as users' sensitivity to discounts or tendency to compare features like discount tags across products. The models cannot compare any specific features between different products. Observations from our eye-tracking Task I support the top-down factors at play. Specifically, users engaged more with the product descriptions, which, despite being less visually attractive, held significant attention in terms of both TTFF (that can also be explained by center bias) and sustained engagement (see Figure 5.3). This engagement with detailed information highlights the importance of user intent, which visual saliency models do not account for. The empirical results from our second eye-tracking experiment provide deeper insights into user attention and engagement in the presence of outliers. The experiments confirmed that outliers capture attention quickly, as evidenced by lower TTFF for outliers compared to distant neighbors. Eye-tracking data also revealed that users engaged with these outliers for longer durations, providing insights into sustained engagement that visual saliency models fail to capture.

In conclusion, our observations suggest that visual saliency models effectively predict initial visual attraction based on basic visual properties. They are useful for quick assessments and designing visually appealing interfaces. However, their reliance on bottom-up factors limits their applicability to fully understand user behavior in real-world scenarios. In contrast, eye-tracking experiments, while more resource-intensive, provide deeper insights into both initial attention and sustained engagement. They capture top-down factors and reflect real-world user interactions more accurately.

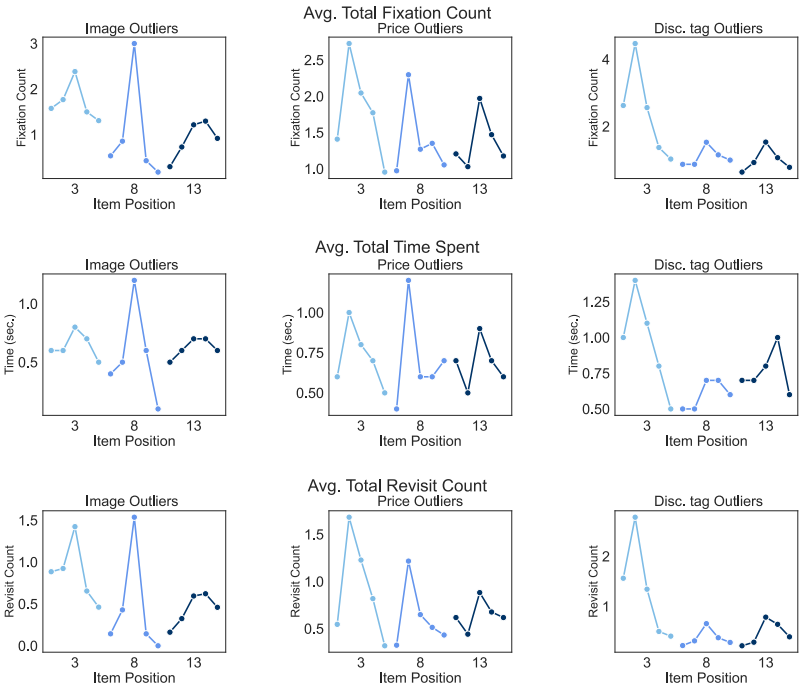


Figure 5.8: Average user engagement metrics across all users, calculated for outliers, their immediate and distant neighbors, across different positions and outlier features.

5.5 Discussion & Conclusion

5.5.1 Research Problem and Objectives

In this study, we explored how different presentational features influence the perception of outliers in e-commerce search results through a two-stage approach. We designed the initial visual search experiments to explore how features such as price, star rating, and discount tags affect users' ability to identify outliers. These experiments provided initial insights into the immediate observability of these features and the impact of visual complexity on user perception.

Building on these preliminary findings, our subsequent eye-tracking experiments aimed to validate and extend our understanding by observing user behavior in a more realistic simulated e-commerce environment. These experiments measured actual user attention and engagement, providing a more comprehensive view of how outlier features capture and sustain attention in real-world-like scenarios. We also incorporated visual saliency analysis to predict which product features would naturally attract attention due to their visual properties. This analysis served as a benchmark to compare with empirical eye-tracking data, allowing us to understand the interplay between bottom-up

visual factors and top-down cognitive factors in shaping user attention during online shopping.

5.5.2 Main Findings

Our initial visual search experiments suggest that visual complexity of a feature affects item outlierness. The visual saliency models confirm this observation by consistently highlighting areas with strong visual contrasts, distinct colors, and complex patterns as attention hotspots. This is expected since these algorithms work based on bottom-up visual factors. Moreover, Figure 5.6 shows that, averaged over all outlier positions, TTFF is the lowest for image outliers, followed by discount tags, and the highest for price outliers ($30.70s$, $35.21s$, and $37.3s$, respectively). Although the product lists used in the eye-tracking experiments were different, the overall trend is in line with our observations from visual search experiments.

Additionally, our observations emphasize that one should be cautious about the limitations of visual saliency models in such contexts. As mentioned, visual saliency models only rely on bottom-up factors, making them naive in that they do not distinguish between separate product features or comparing them against each other, instead they analyze the entire image and highlight areas that stand out based on overall visual complexity. Therefore, in lists with colorful and complex product images, the models might miss an obvious outlier such as a unique discount tag, while detecting a subtle visual difference made by a higher price in a list with uniform dark colors.

Moreover, our eye-tracking Task I suggests that despite being less visually attractive, product descriptions captured attention more quickly, indicating the importance of the top-down factors and other factors in play like center bias. This is evident by a lower TTFF for product descriptions in Figure 5.3a followed by more revisits and time spent on these areas, reflecting deeper cognitive engagement (see Figures 5.3b and 5.3c).

In Task II of our eye-tracking study, we examined how outlier product features influence visibility in search result lists. The results indicated that outliers and their immediate neighbors attracted attention faster (in terms of TTFF) and engaged users for longer durations (in terms of fixation count and time spent) compared to distant items. This effect was consistent across different outlier features (image, price, discount tag) and various positions within the list.

Overall, our findings from visual search, visual saliency models, and eye-tracking experiments emphasize the dual role of visual and cognitive factors in shaping user attention. Visual saliency models effectively predict initial visual attraction based on bottom-up factors, while eye-tracking data provides comprehensive insights into sustained engagement driven by the top-down factors. These insights can inform the design of more effective and engaging e-commerce interfaces by optimizing the presentation of key product features to capture and maintain user attention.

5.A Task Instructions

In the following, we provide the instructions that were shared with participants on the Prolific platform for our experiments.

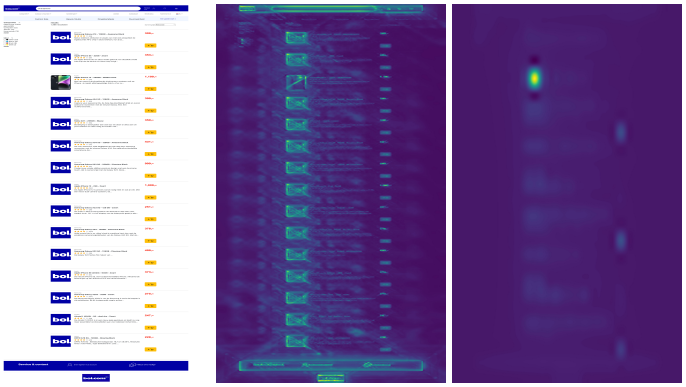
In this task, you'll review some product lists on an online shopping website (Bol.com). You'll explore a specific category, like smartphones, shoes, or backpacks, as if you're planning to make a purchase.

We require **access to your webcam** to record your **eye movements** as you review the product lists. Please be assured that we will only record eye movements on the product list pages, and your personal or private data will not be recorded or accessed in any way.

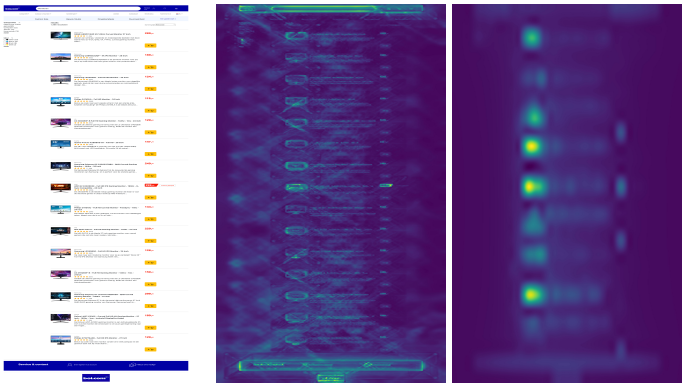
Please carefully examine all the products on the page. **We'll ask questions about your observations afterward**, such as:

- Describe what you saw on the list briefly, e.g., price range, item types.
- Note if you observed a specific gender focus or prominent colors.
- List any brands you noticed.
- Recommend one item for purchase and explain why.
- Mention anything that caught your attention.

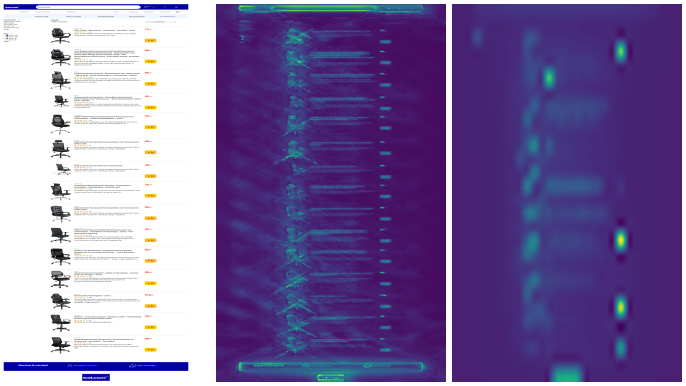
Your answers can be in English or Dutch. **IMPORTANT**: You **MUST** answer the post-task questions about the content of the lists accurately, otherwise we **CANNOT ACCEPT** your submission.



(a) (b) (c)

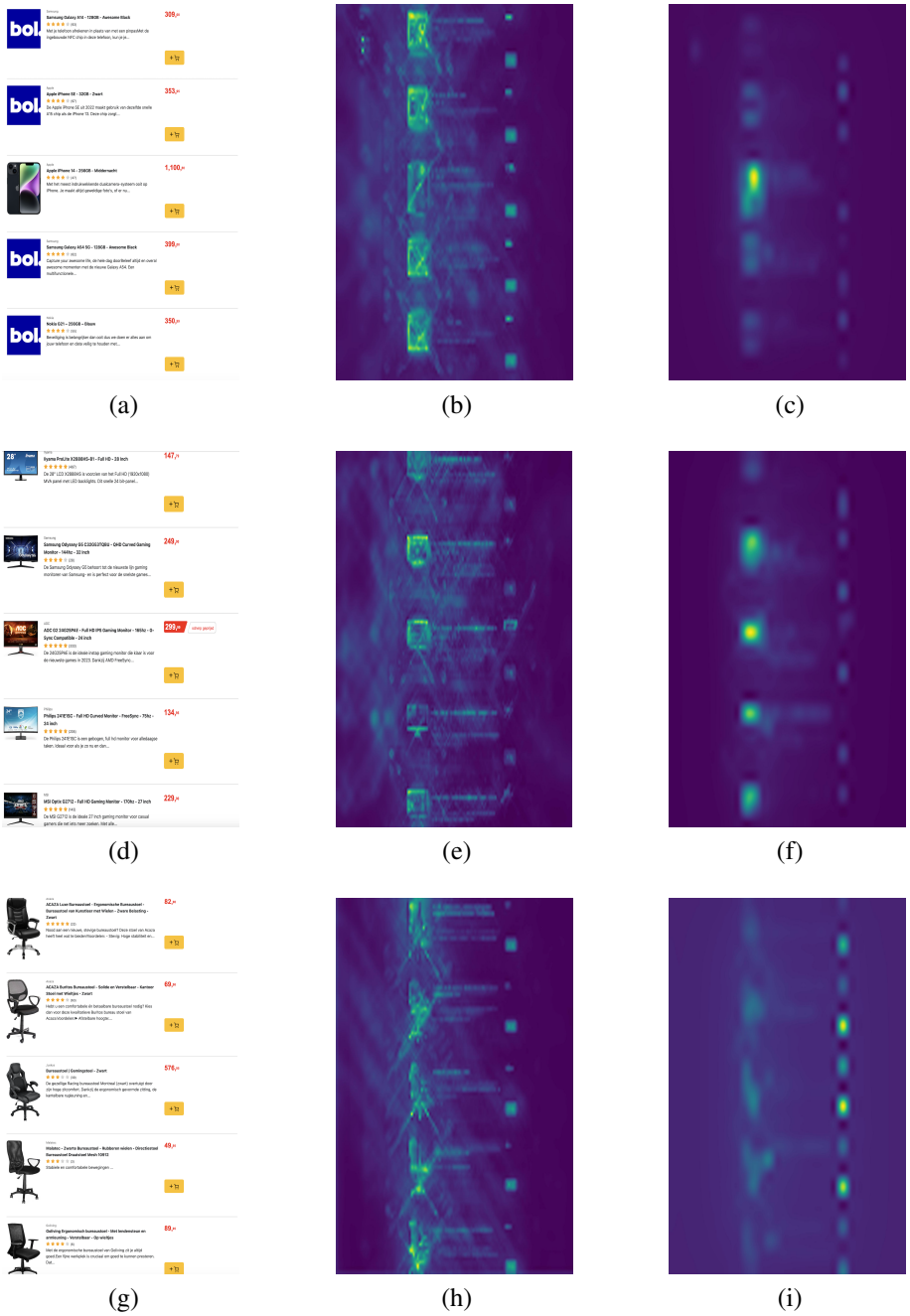


(d) (e) (f)



(g) (h) (i)

Figure 5.9: Product lists and their visual saliency maps. (a), (d) and (g) show the original list for mobile phones with an outlier image at position 3, monitors with an outlier discount tag at position 8, and office chairs with an outlier price at position 13. (b), (e) and (h) show the corresponding visual saliency maps using the Itti & Koch model. (c), (f), and (i) show the maps generated by the GBVS.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 5.10: Product lists and their visual saliency maps. (a), (d) and (g) show the original list with one outlier item and its close neighbors: for mobile phones at position 3, for monitors at position 8 and for office chairs at position 13, respectively. (b), (e) and (h) show the visual saliency maps generated by the Itti & Koch model, while (c), (f) and (i) are generated by the GBVS.

6 Conclusions

In this thesis, we have examined several specific challenges in applying learning to rank (LTR) models to product search. Specifically, we have covered (i) a comprehensive evaluation of 12 supervised learning-to-match methods, offering insights for selecting methods that balance effectiveness and efficiency in real-world applications; and (ii) the concept of inter-item dependencies in fair ranking, introducing "outlierness" and examining its impact on exposure distribution and fairness, along with a method to mitigate these effects. Subsequently, we have shifted our focus to user examination behavior in the presence of outliers, where we (iii) have introduced a new type of click bias, i.e., outlier bias and proposed a click model that accounts for both outlier and position bias. Finally, we have investigated (iv) how different presentational features impact the perception of outliers in e-commerce search, using visual saliency models and eye-tracking experiments. In this chapter, we review our main findings and suggest directions for future work.

6.1 Main Findings

In this section, we revisit the research questions that were posed in Chapter 1 and summarize the most important findings.

RQ1 How do learning-to-match models perform in ranking for product search compared to each other in terms of efficiency and accuracy?

Through a systematic comparison of 12 supervised learning-to-match methods, we found that models that have been specifically designed for short text matching, such as MV-LSTM and DRMMTKS, are consistently among the top performing methods in all experiments; however, ARC-I is the preferred model for real world use cases, when taking efficiency and accuracy into account at the same time. Moreover, although the state-of-the-art BERT-based model is the fastest in both training and inference, its performance is only mediocre compared to other models. We attribute this result to the fact that the text BERT is pre-trained on is very different from the text we have in product search. Lastly, we have provided insights into factors that can influence model behavior for different types of query, such as the length of the retrieved list, and query complexity. We have also discussed the implications of our findings for e-commerce practitioners, with respect to choosing a well performing method.

RQ2 Do outlier items exist in search logs, and how can their effect on exposure-based fair ranking algorithms be mitigated?

To answer the first part of this question, we have analyzed data from the TREC Fair Ranking track and found a significant number of outliers in the rankings. Next, to confirm our hypothesis about the impact of outlier items on users' scanning behavior, we have conducted an eye-tracking study in two scenarios: e-commerce and scholarly search. To account for the existence and effect of outliers in a ranked list we have proposed OMIT. With OMIT, we have introduced a ranking constraint based on the outlierness of items in a list and combined it with fairness constraints. OMIT can reduce outliers in rankings without compromising user utility or position-based item fairness. Through experiments on a public dataset, we have shown that outlierness optimization leads to a fairer policy that displays fewer outliers in the top results, while maintaining a reasonable trade-off between fairness and utility.

RQ3 Does outlier bias exist in click data? How can we estimate its impact and correct for this bias?

We have empirically shown that, on average, outlier items receive significantly more clicks than non-outlier items in the same lists. This tendency holds across all positions; at any specific rank, an item receives more interactions when presented as an outlier rather than a non-outlier item. We concluded from our analysis that the effect of outliers on user clicks is a type of bias that distorts true relevance. Therefore, we have proposed OPBM, an outlier-aware click model that accounts for both outlier and position bias. We estimate click propensities based on OPBM and validate our model's effectiveness through extensive experiments on both real-world e-commerce data and semi-synthetic datasets. Our results demonstrate that OPBM outperforms baseline models in ranking performance and true relevance estimation. We have also shown that OPBM performs similarly to PBM in the worst case, making it a more reliable choice.

RQ4 How do different presentational features shape users' perception of outliers and influence exposure distribution in e-commerce search results?

We have investigated the impact of different presentational features on users' perception of outlierness by first, modeling the problem as a visual search task. We have found that participants perceive each presentational feature differently in terms of attention and reaction times. To better examine the factors involved, we have categorized them into two established groups: bottom-up and top-down. We started by analyzing the influence of bottom-up factors through experiments with visual saliency models. We show that graph-based visual saliency model is effective in detecting visual anomalies in ranked lists, while the well-known Itti & Koch model detects general visual attention patterns in an image. Next, through eye-tracking experiments, we have demonstrated the impact of top-down factors on user attention, showing that these factors can override bottom-up visual signals in online shopping scenarios. We have found that outlier items and their close neighbors in ranked lists not only attract attention more quickly but also receive significantly more exposure, as users spend more time examining them.

6.2 Future Work

In this thesis, we have presented several analyses and solutions for applying LTR models effectively to product search, addressing query-product matching, unbiased learning to rank , and fair ranking. Specifically, we have analyzed existing methods for supervised learning-to-match to assess their applicability in real-world e-commerce search (Chapter 2). In the next chapters, we extend current approaches by refining LTR methods to account for inter-item dependencies (Chapters 3 and 4) and examining the impact of different product presentational features on item outlierness (Chapter 5). In this section, we outline limitations and suggest future directions for further development of these topics.

6.2.1 Learning to Match

In Chapter 2 we have compared 12 learning-to-match models for product search using two datasets: one public, hashed dataset and one private dataset from an e-commerce platform. We observed performance differences across these datasets, which raises questions about the generalizability of some models. Validating these observations with additional unhashed datasets, such as [130], could help address this limitation. Additionally, with rapid advancements in semantic matching, including the rise of large language models and retrieval-augmented generation approaches [78, 192], replicating our experiments with more recent models would provide further insights.

6.2.2 Outliers

Our proposed model, OMIT, for mitigating outlierness in ranked lists focuses on removing outliers from the top-k positions. These outliers are defined in the context of the entire list. Improving this model to mitigate outlierness across all sliding windows of size k could be an interesting future direction. This adjustment would bring the model closer to real-world conditions. Furthermore, in Chapter 3, we observe that most of the outliers in our dataset are irrelevant items. Therefore, removing these outliers from the top-k positions does not have a large negative effect on utility. Future work could focus on improving performance in cases where outliers are relevant items, e.g., by considering alternative methods of outlier mitigation.

In Chapter 4 we have introduced outlier bias and proposed a click model, OPBM, that accounts for both position and outlier bias. We have discussed how OPBM generalizes to cases with multiple outliers in rankings through experiments with semi-synthetic data. One limitation of our work, in that chapter, is that for rankings with multiple outliers, we have assumed that the effect of each outlier is independent of its position and the presence of other outliers. One direction for future work, therefore, is to investigate how multiple outliers on the same ranking affect each other and their surrounding items. Another interesting extension of our work is to study how outlier bias can compensate for position bias in the top-k ranks, and explore its use in different domains such as fairness of exposure.

In the final chapter of this thesis, we have provided insights into how different features contribute to an item's outlierness. However, our analyses do not directly

estimate the exposure of outlier features, which depends on various user behaviors and platform-specific algorithms outside the scope of this study. Future work should focus on quantifying and generalizing the impact of different visual attributes on item outlierness to understand broader implications. For generalizability, we focused on presentational features that are common across e-commerce platforms, but item presentation can vary significantly among different interfaces. Additionally, our study is limited to list-view presentations; findings may differ in grid views or other layouts. Extending these analyses to various layouts could provide further insights. Another natural extension of this research is to use eye-tracking data as training inputs for visual saliency models, aiming to create models that predict attention areas specifically for e-commerce. To our knowledge, no existing methods combine both top-down and bottom-up factors for visual saliency prediction in e-commerce.

6.2.3 Zooming Out

In this thesis we have studied several challenges involved in applying LTR methods for product search. These insights can serve as a foundation for addressing more domain-specific ranking issues. For instance, recent advances in language modeling can help address query-product matching. One of the major challenges in e-commerce search is the scarcity of open-source, high-quality datasets [20, 121]. By using generative models, such as large language models, researchers can create synthetic datasets that mimic the structure and diversity of real e-commerce data [132, 133, 193], capturing the short, unstructured, and domain-specific nature of product titles, descriptions and user queries. This approach can also support the pre-training of transformer-based models with data better suited to e-commerce contexts.

Additionally, future work could explore a broader range of inter-item dependencies beyond outliers. This requires examining different types of dependencies, such as how items influence the visibility, appeal, or perceived relevance of their neighbors within ranked lists. Understanding these dependencies not only allows for mitigating bias in the data but also offers opportunities to actively leverage them to achieve targeted exposure distribution goals. For example, exposure distributions can be optimized to promote fairness in two-sided marketplaces [24, 71], or even support sustainability goals by prioritizing items with a shorter shelf life in grocery shopping to reduce waste [75]. By tailoring exposure distribution according to these varied goals, inter-item dependency modeling could contribute to more balanced and purpose-driven ranking strategies. Furthermore, another future direction concerns investigating the potential to apply this research in cross-domain contexts, such as media or news ranking, where both exposure fairness and relevance are critical.

Furthermore, instead of treating ranking fairness as a post-processing step, future work could integrate fairness constraints directly into learning-to-match models, optimizing for both relevance and balanced exposure. This could be achieved through a multi-objective ranking framework that simultaneously accounts for semantic relevance, presentation effects, and fairness constraints.

Bibliography

[1] H. Abdollahpouri, R. Burke, and B. Mobasher. Controlling popularity bias in learning-to-rank recommendation. In *RecSys*, pages 42–46, 2017. (Cited on pages 47, 52, and 68.)

[2] A. Agarwal, X. Wang, C. Li, M. Bendersky, and M. Najork. Addressing trust bias for unbiased learning-to-rank. In *WWW*, pages 4–14, 2019. (Cited on pages 47, 68, and 72.)

[3] A. Agarwal, I. Zaitsev, X. Wang, C. Li, M. Najork, and T. Joachims. Estimating position bias without intrusive interventions. In *WSDM*, pages 474–482, 2019. (Cited on pages 52 and 69.)

[4] P. Aggarwal and R. Vaidyanathan. Is font size a big deal? A transaction–acquisition utility perspective on comparative price promotions. *Journal of Consumer Marketing*, 2016. (Cited on pages 56 and 73.)

[5] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. Diversifying search results. In *WSDM*, page 5–14, 2009. (Cited on page 56.)

[6] Q. Ai, K. Bi, C. Luo, J. Guo, and W. B. Croft. Unbiased learning to rank with unbiased propensity estimation. In *SIGIR*, pages 385–394, 2018. (Cited on pages 53, 62, and 69.)

[7] Q. Ai, T. Yang, H. Wang, and J. Mao. Unbiased learning to rank: Online or offline? *ACM Transactions on Information Systems (TOIS)*, 39(2):1–29, 2021. (Cited on page 69.)

[8] B. Albert and T. Tullis. *Measuring the user experience: collecting, analyzing, and presenting usability metrics*. Newnes, 2013. (Cited on page 84.)

[9] C. Andrade. Internal, external, and ecological validity in research design, conduct, and evaluation. *Indian journal of psychological medicine*, 40(5):498–499, 2018. (Cited on pages 3 and 54.)

[10] F. Angiulli and C. Pizzuti. Fast outlier detection in high dimensional spaces. In *ECML PKDD*, pages 15–27, 2002. (Cited on page 34.)

[11] A. Antoniades, X. Wang, Y. Elazar, A. Amayuelas, A. Albalak, K. Zhang, and W. Y. Wang. Generalization vs. memorization: Tracing language models' capabilities back to pretraining data. In *ICML 2024 Workshop on Foundation Models in the Wild*, 2024. (Cited on page 29.)

[12] I. Arapakis, X. Bai, and B. B. Cambazoglu. Impact of response latency on user behavior in web search. In *SIGIR*, pages 103–112. ACM, 2014. (Cited on page 16.)

[13] D. Ariely and G. S. Berns. Neuromarketing: The hope and hype of neuroimaging in business. *Nature Reviews Neuroscience*, 11(4):284–292, 2010. (Cited on page 82.)

[14] A. Asudeh, H. Jagadish, J. Stoyanovich, and G. Das. Designing fair ranking schemes. In *SIGMOD*, pages 1259–1276, 2019. (Cited on page 47.)

[15] L. Azzopardi. Cognitive biases in search: A review and reflection of cognitive biases in information retrieval. In *CHIIR*, pages 27–37. ACM, 2021. (Cited on page 79.)

[16] R. Baeza-Yates. Bias on the web. *Communications of the ACM*, 61(6):54–61, 2018. (Cited on pages 2, 3, 31, and 34.)

[17] R. Baeza-Yates, B. Ribeiro-Neto, et al. *Modern information retrieval*, volume 463. ACM press New York, 1999. (Cited on page 1.)

[18] A. Bell, P. Senthil Kumar, and D. Miranda. The title says it all: A title term weighting strategy for ecommerce ranking. In *CIKM*, pages 2233–2241, 2018. (Cited on page 11.)

[19] J. S. Bergstra, Bardenet, and Rémi. Algorithms for hyper-parameter optimization. In *NeurIPS*, pages 2546–2554, 2011. (Cited on page 17.)

[20] A. Berke, D. Calacci, R. Mahari, T. Yabe, K. Larson, and S. Pentland. Open e-commerce 1.0, five years of crowdsourced us Amazon purchase histories with user demographics. *Scientific Data*, 11(1): 491, 2024. (Cited on page 102.)

[21] A. J. Biega, K. P. Gummadi, and G. Weikum. Equity of attention: Amortizing individual fairness in rankings. In *SIGIR*, pages 405–414, 2018. (Cited on pages 3, 31, 34, 35, 48, 65, and 72.)

[22] A. J. Biega, F. Diaz, M. D. Ekstrand, and S. Kohlmeier. Overview of the TREC 2019 fair ranking track. *arXiv preprint arXiv:2003.11650*, 2020. (Cited on page 3.)

[23] G. Birkhoff. *Lattice Theory*. AMS, 1940. (Cited on pages 33, 41, and 42.)

[24] A. Biswas, G. K. Patro, N. Ganguly, K. P. Gummadi, and A. Chakraborty. Toward fair recommendation in two-sided platforms. *ACM Transactions on the Web (TWEB)*, 16(2):1–34, 2021. (Cited on page 102.)

[25] A. Borisov, I. Markov, M. de Rijke, and P. Serdyukov. A neural click model for web search. In *WWW*, pages 531–541, 2016. (Cited on page 32.)

[26] S. Brandl, O. Eberle, T. Ribeiro, A. Søgaard, and N. Hollenstein. Evaluating webcam-based gaze data as an alternative for human rationale annotations. *arXiv preprint arXiv:2402.19133*, 2024. (Cited on page 84.)

[27] E. Brenner, J. Zhao, et al. End-to-end neural ranking for ecommerce product search. In *eCom: The SIGIR 2018 Workshop on eCommerce*, 2018. (Cited on page 12.)

[28] G. T. Buswell. *How People Look at Pictures: A Study of the Psychology and Perception in Art*. Univ. Chicago Press, 1935. (Cited on pages 74 and 87.)
[29] Z. Cao, T. Qin, T. Liu, M. Tsai, and H. Li. Learning to rank: From pairwise approach to listwise approach. In *ICML*, pages 129–136, 2007. (Cited on page 44.)
[30] L. E. Celis, D. Straszak, and N. K. Vishnoi. Ranking with fairness constraints. In *ICALP*, pages 28:1–28:15, 2018. (Cited on page 47.)
[31] L. E. Celis, A. Mehrotra, and N. K. Vishnoi. Interventions for ranking in the presence of implicit bias. In *FAT**, pages 369–380, 2020. (Cited on page 47.)
[32] O. Chapelle and Y. Chang. Yahoo! Learning to Rank Challenge Overview. *Journal of Machine Learning Research*, 14:1–24, 2011. (Cited on page 62.)
[33] R.-C. Chen, Q. Ai, G. Jayasinghe, and W. B. Croft. Correcting for recency bias in job recommendation. In *CIKM*, pages 2185–2188, 2019. (Cited on pages 47, 52, and 68.)
[34] K.-P. Chiang and R. R. Dholakia. Factors driving consumer intention to shop online: an empirical investigation. *Journal of Consumer Psychology*, 13(1-2):177–183, 2003. (Cited on page 83.)
[35] R. Chocarro Eguaras, M. Cortiñas Ugalde, and A. Villanueva Larre. Attention to product images in an online retailing store: An eye-tracking study considering consumer goals and type of product. *Journal of Electronic Commerce Research 23 (4), 257-281*, 2022. (Cited on pages 81, 82, and 83.)
[36] A. Chuklin, I. Markov, and M. de Rijke. *Click Models for Web Search*. Synthesis Lectures on Information Concepts, Retrieval, and Services. Morgan & Claypool Publishers, August 2015. (Cited on pages 32, 47, 60, and 68.)
[37] N. Craswell, O. Zoeter, M. Taylor, and B. Ramsey. An experimental comparison of click position-bias models. In *WSDM*, pages 87–94, 2008. (Cited on page 17.)
[38] Y. Cui, F. Liu, P. Wang, B. Wang, H. Tang, Y. Wan, J. Wang, and J. Chen. Distillation matters: empowering sequential recommenders to match the performance of large language models. In *RecSys*, pages 507–517, 2024. (Cited on page 29.)
[39] Z. Dai, C. Xiong, et al. Convolutional neural networks for soft-matching n-grams in ad-hoc search. In *WSDM*, pages 126–134. ACM, 2018. (Cited on page 14.)
[40] W. de Vries, A. van Cranenburgh, et al. Bertje: A Dutch BERT model. *arXiv preprint arXiv:1912.09582*, 2019. (Cited on page 17.)
[41] J. Devlin, M.-W. Chang, et al. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. (Cited on pages 14 and 28.)
[42] F. Diaz, B. Mitra, M. D. Ekstrand, A. J. Biega, and B. Carterette. Evaluating stochastic rankings with expected exposure. In *CIKM*, pages 275–284, 2020. (Cited on pages 48 and 72.)
[43] H. Duan, C. Zhai, et al. Supporting keyword search in product database: A probabilistic approach. *VLDB*, 6(14):1786–1797, 2013. (Cited on page 11.)
[44] J. Duncan and G. W. Humphreys. Visual search and stimulus similarity. *Psychological Review*, 96(3): 433, 1989. (Cited on pages 73, 76, 79, and 80.)
[45] D. EStimator. A fast algorithm for the minimum covariance. *Technometrics*, 41(3):212, 1999. (Cited on page 75.)
[46] Z. Fang, A. Agarwal, and T. Joachims. Intervention harvesting for context-dependent examination-bias estimation. In *SIGIR*, pages 825–834, 2019. (Cited on page 60.)
[47] M. Fazio, A. Reitano, and M. R. Loizzo. Consumer preferences for new products: Eye tracking experiment on labels and packaging for olive oil based dressing. In *Proceedings*, volume 70. MDPI, 2020. (Cited on page 84.)
[48] G. Federico and M. A. Brandimonte. Tool and object affordances: An ecological eye-tracking study. *Brain and Cognition*, 135:103582, 2019. (Cited on page 84.)
[49] S. Fiedler, M. Schulte-Mecklenbeck, F. Renkewitz, and J. L. Orquin. Guideline for reporting standards of eye-tracking research in decision sciences. *PsyArXiv preprint*, 2020. (Cited on pages 38 and 84.)
[50] T. Foulsham and G. Underwood. What can saliency models predict about eye movements? spatial and sequential aspects of fixations during encoding and recognition. *Journal of Vision*, 8(2):6–6, 2008. (Cited on pages 74 and 87.)
[51] S. C. Geyik, S. Ambler, and K. Kenthapadi. Fairness-aware ranking in search & recommendation systems with application to linkedin talent search. In *SIGKDD*, pages 2221–2231, 2019. (Cited on page 47.)
[52] L. Giovannangeli, R. Bourqui, R. Giot, and D. Auber. Color and shape efficiency for outlier detection from automated to user evaluation. *Visual Informatics*, 2022. (Cited on pages 73 and 76.)
[53] J. Guo, Y. Fan, et al. A deep relevance matching model for ad-hoc retrieval. In *CIKM*, pages 55–64. ACM, 2016. (Cited on pages 12 and 13.)

[54] J. Guo, Y. Fan, et al. A deep look into neural ranking models for information retrieval. *IPM*, page Article 102067, 2019. (Cited on pages 12, 14, 24, and 25.)
[55] J. Guo, Y. Fan, et al. Matchzoo: A learning, practicing, and developing system for neural text matching. *arXiv preprint arXiv:1905.10289*, 2019. (Cited on pages 12, 13, and 17.)
[56] T. Guo, T. Lin, and Y. Lu. An interpretable LSTM neural network for autoregressive exogenous model. *arXiv preprint arXiv:1804.05251*, 2018. (Cited on page 13.)
[57] M. Haldar, H. Zhang, K. Bellare, S. Chen, S. Banerjee, X. Wang, M. Abdool, H. Gao, P. Tapadia, L. He, et al. Learning to rank for maps at airbnb. *arXiv preprint arXiv:2407.00091*, 2024. (Cited on page 84.)
[58] J. Harel, C. Koch, and P. Perona. Graph-based visual saliency. *Advances in Neural Information Processing Systems*, 19, 2006. (Cited on pages 74 and 82.)
[59] M. Heuss, F. Sarvi, and M. de Rijke. Fairness of exposure in light of incomplete exposure estimation. In *SIGIR*, pages 759–769, 2022. (Cited on page 65.)
[60] K. Hofmann, A. Schuth, A. Bellogin, and M. de Rijke. Effects of position bias on click-based recommender evaluation. In *ECIR*, pages 624–630. Springer, 2014. (Cited on page 24.)
[61] B. Hu, Z. Lu, et al. Convolutional neural network architectures for matching natural language sentences. In *NIPS*, pages 2042–2050, 2014. (Cited on page 13.)
[62] L. Huang, C.-H. Tan, W. Ke, and K.-K. Wei. Do we order product review information display? how? *Information & Management*, 51(7):883–894, 2014. (Cited on page 83.)
[63] P.-S. Huang, X. He, et al. Learning deep structured semantic models for web search using clickthrough data. In *CIKM*, pages 2333–2338. ACM, 2013. (Cited on pages 12 and 14.)
[64] B. Iglewicz and D. C. Hoaglin. *How to Detect and Handle Outliers*. ASQ Quality Press, 1993. (Cited on page 34.)
[65] L. Itti. Visual salience. *Scholarpedia*, 2(9):3327, 2007. (Cited on page 75.)
[66] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, 1998. (Cited on pages 74, 75, and 82.)
[67] R. Jagerman, H. Oosterhuis, and M. de Rijke. To model or to intervene: A comparison of counterfactual and online learning to rank from user interactions. In *SIGIR*, pages 15–24, 2019. (Cited on pages 62 and 63.)
[68] W. Jin, A. K. Tung, J. Han, and W. Wang. Ranking outliers using symmetric neighborhood relationship. In *PAKDD*, pages 577–593. Springer, 2006. (Cited on page 75.)
[69] W. Jin, A. K. H. Tung, J. Han, and W. Wang. Ranking outliers using symmetric neighborhood relationship. In *PAKDD*, pages 577–593, 2006. (Cited on pages 34 and 68.)
[70] T. Joachims. Optimizing search engines using clickthrough data. In *KDD*, pages 133–142, 2002. (Cited on page 17.)
[71] T. Joachims. Fairness and control of exposure in two-sided markets. In *SIGIR*, pages 1–1, 2021. (Cited on page 102.)
[72] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. In *SIGIR*, pages 154–161, 2005. (Cited on pages 2, 3, 31, 32, 37, 47, 52, 68, and 72.)
[73] T. Joachims, L. Granka, B. Pan, H. Hembrooke, F. Radlinski, and G. Gay. Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. *ACM Transactions on Information Systems (TOIS)*, 25(2):7–es, 2007. (Cited on pages 51 and 60.)
[74] T. Joachims, A. Swaminathan, and T. Schnabel. Unbiased learning-to-rank with biased feedback. In *WSDM*, pages 781–789, 2017. (Cited on pages 47, 52, 53, 60, 62, 63, 68, 69, and 72.)
[75] S. Jullien, M. Ariannezhad, P. Groth, and M. de Rijke. A simulation environment and reinforcement learning method for waste reduction. *arXiv preprint arXiv:2205.15455*, 2022. (Cited on page 102.)
[76] K. C. Kao, S. R. Hill, and I. Troshani. Effects of cue congruence and perceived cue authenticity in online group buying. *Internet Research*, 2020. (Cited on pages 56 and 73.)
[77] S. K. Karmaker Santu, P. Sondhi, and C. Zhai. On application of learning to rank for e-commerce search. In *SIGIR*, pages 475–484, 2017. (Cited on pages 1 and 11.)
[78] O. Khattab and M. Zaharia. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *SIGIR*, pages 39–48, 2020. (Cited on pages 28 and 101.)
[79] M. Kim and S. Lennon. The effects of visual and verbal information on attitudes and purchase intentions in internet shopping. *Psychology & Marketing*, 25(2):146–178, 2008. (Cited on page 83.)
[80] A. Kittur, E. H. Chi, and B. Suh. Crowdsourcing user studies with mechanical turk. In *CHI*, pages 453–456. ACM, 2008. (Cited on page 77.)

[81] D. Lee and K. Hosanagar. How do product attributes and reviews moderate the impact of recommender systems through purchase stages? *Management Science*, 67(1):524–546, 2021. (Cited on page 83.)
[82] L. A. Leiva, Y. Xue, A. Bansal, H. R. Tavakoli, T. Köroðlu, J. Du, N. R. Dayama, and A. Oulasvirta. Understanding visual saliency in mobile user interfaces. In *22nd International Conference on Human-Computer Interaction with Mobile Devices and Services*, pages 1–12, 2020. (Cited on pages 75 and 82.)
[83] A. M. Levin, I. R. Levin, and C. E. Heath. Product category dependent consumer preferences for online and offline shopping features and their influence on multi-channel retail alliances. *Journal of Electronic Commerce Research*, 4(3):85–93, 2003. (Cited on page 83.)
[84] D. Lewandowski and Y. Kammerer. Factors influencing viewing behaviour on search engine results pages: A review of eye-tracking research. *Behaviour & Information Technology*, 40(14):1485–1515, 2021. (Cited on page 85.)
[85] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474, 2020. (Cited on page 28.)
[86] D. J. Lewkowicz. The concept of ecological validity: What are its limitations and is it bad to be invalid? *Infancy*, 2(4):437–450, 2001. (Cited on pages 3 and 54.)
[87] H. Li and J. Xu. Semantic matching in search. *FnTIR*, 7(5):343–469, 2014. (Cited on page 10.)
[88] Y. Li, S. Ma, X. Wang, S. Huang, C. Jiang, H.-T. Zheng, P. Xie, F. Huang, and Y. Jiang. Ecomgpt: Instruction-tuning large language models with chain-of-task tasks for e-commerce. In *AAAI*, pages 18582–18590, 2024. (Cited on page 29.)
[89] Z. Li, Y. Zhao, N. Botta, C. Ionescu, and X. Hu. COPOD: copula-based outlier detection. In *2020 IEEE international conference on data mining (ICDM)*, pages 1118–1123. IEEE, 2020. (Cited on pages 35, 68, and 75.)
[90] E. Liberty, Z. Karnin, et al. Elastic machine learning algorithms in Amazon sagemaker. In *SIGMOD*, pages 731–737, 2020. (Cited on page 16.)
[91] T. Linjordet and K. Balog. Impact of training dataset size on neural answer selection models. In *ECIR*, pages 828–835. Springer, 2019. (Cited on page 12.)
[92] T.-Y. Liu et al. Learning to rank for information retrieval. *FnTIR*, 3(3):225–331, 2009. (Cited on page 1.)
[93] S. Lu and J.-H. Lim. Saliency modeling from image histograms. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VII 12*, pages 321–332. Springer, 2012. (Cited on page 75.)
[94] J. Luan, Z. Yao, F. Zhao, and H. Liu. Search product and experience product online reviews: An eye-tracking study on consumers' review search behavior. *Computers in Human Behavior*, 65:420–430, 2016. (Cited on page 83.)
[95] M. Ludewig and D. Jannach. Learning to rank hotels for search and recommendation from session-based interaction logs and meta data. In *RecSys Challenge '19*. ACM, 2019. (Cited on page 11.)
[96] M. Ludewig, N. Mauro, et al. Performance comparison of neural and non-neural approaches to session-based recommendation. In *RecSys*, RecSys '19, pages 462–466. ACM, 2019. (Cited on page 12.)
[97] S. MacAvaney, A. Yates, et al. CEDR: Contextualized embeddings for document ranking. In *SIGIR*, pages 1101–1104, 2019. (Cited on pages 14, 15, and 17.)
[98] A. Magnani, F. Liu, et al. Neural product retrieval at walmart.com. In *WWW*, pages 367–372, 2019. (Cited on page 12.)
[99] N. M. Majeed, Y. J. Chua, M. Kothari, M. Kaur, F. Y. Quek, M. H. Ng, W. Q. Ng, and A. Hartanto. Anxiety disorders and executive functions: A three-level meta-analysis of reaction time and accuracy. *Psychiatry Research Communications*, 3(1):100100, 2023. (Cited on pages 75 and 81.)
[100] M. Marcus and R. Ree. Diagonals of doubly stochastic matrices. *The Quarterly Journal of Mathematics*, 10(1):296–302, 1959. (Cited on page 42.)
[101] B. McElree and M. Carrasco. The temporal dynamics of visual search: Evidence for parallel processing in feature and conjunction searches. *Journal of Experimental Psychology: Human Perception and Performance*, 25(6):1517, 1999. (Cited on page 76.)
[102] P. E. McKight and J. Najab. Kruskal-Wallis test. *The Corsini Encyclopedia of Psychology*, pages 1–1, 2010. (Cited on page 87.)
[103] R. Mehrotra, J. McInerney, H. Bouchard, M. Lalmas, and F. Diaz. Towards a fair marketplace: Counterfactual evaluation of the trade-off between relevance, fairness & satisfaction in recommendation systems. In *CIKM*, pages 2243–2251, 2018. (Cited on pages 3, 31, 35, 48, and 72.)

[104] P. Metrikov, F. Diaz, S. Lahaie, and J. Rao. Whole page optimization: How page elements interact with the position auction. In *EC*, pages 583–600, 2014. (Cited on pages 47 and 68.)
[105] B. Mitra and N. Craswell. Neural models for information retrieval. *arXiv preprint arXiv:1705.01509*, 2017. (Cited on pages 1, 3, 10, and 11.)
[106] B. Mitra, F. Diaz, and N. Craswell. Learning to match using local and distributed representations of text for web search. In *WWW*, pages 1291–1299. ACM, 2017. (Cited on page 14.)
[107] M. Morik, A. Singh, J. Hong, and T. Joachims. Controlling fairness and bias in dynamic learning-to-rank. In *SIGIR*, pages 429–438, 2020. (Cited on pages 3, 31, 35, 48, 65, and 72.)
[108] G. Müller-Plath and S. Pollmann. Determining subprocesses of visual feature search with reaction time models. *Psychological Research*, 67:80–105, 2003. (Cited on pages 75 and 81.)
[109] M. Murali and A. Çöltekin. Conducting eye tracking studies online. In *Workshop on Adaptable Research Methods for Empirical Research with Map Users, Virtual Workshop*, volume 6, 2021. (Cited on page 84.)
[110] P. Nelson. Information and consumer behavior. *Journal of Political Economy*, 78(2):311–329, 1970. (Cited on page 83.)
[111] R. Nogueira, Z. Jiang, and J. Lin. Document ranking with a pretrained sequence-to-sequence model. *arXiv preprint arXiv:2003.06713*, 2020. (Cited on page 28.)
[112] A. Novin and E. M. Meyers. Making sense of conflicting science information: Exploring bias in the search engine result page. In *CHIIR*, pages 175–184. ACM, 2017. (Cited on page 72.)
[113] M. O'Brien and M. T. Keane. Modeling result-list searching in the world wide web: The role of relevance topologies and trust bias. In *CogSci*, pages 1881–1886, 2006. (Cited on page 32.)
[114] K. D. Onal, Y. Zhang, I. S. Altingovde, M. M. Rahman, P. Karagoz, A. Braylan, B. Dang, H.-L. Chang, H. Kim, Q. Mcnamara, et al. Neural information retrieval: at the end of the early years. *Information Retrieval*, 21(2-3):111–182, 2018. (Cited on pages 1, 3, 10, and 11.)
[115] H. Oosterhuis and M. de Rijke. Differentiable unbiased online learning to rank. In *CIKM*, pages 1293–1302, 2018. (Cited on page 69.)
[116] H. Oosterhuis and M. de Rijke. Policy-aware unbiased learning to rank for top-k rankings. In *SIGIR*, pages 489–498, 2020. (Cited on pages 53, 62, and 63.)
[117] H. Oosterhuis and M. de Rijke. Unifying online and counterfactual learning to rank: A novel counterfactual estimator that effectively utilizes online interventions. In *WSDM*, pages 463–471, 2021. (Cited on page 32.)
[118] Z. Ovaisi, R. Ahsan, Y. Zhang, K. Vasilaky, and E. Zheleva. Correcting for selection bias in learning-to-rank systems. In *WWW*, pages 1863–1873, 2020. (Cited on pages 47, 52, 68, and 72.)
[119] E. M. Palmer, T. S. Horowitz, A. Torralba, and J. M. Wolfe. What are the shapes of response time distributions in visual search? *Journal of Experimental Psychology: Human Perception and Performance*, 37(1):58, 2011. (Cited on pages 75 and 81.)
[120] L. Pang, Y. Lan, et al. Text matching as image recognition. In *AAAI*, 2016. (Cited on page 14.)
[121] A. Papenmeier, D. Kern, D. Hienert, A. Sliwa, A. Aker, and N. Fuhr. Dataset of natural language queries for e-commerce. In *CHIIR*, pages 307–311, 2021. (Cited on page 102.)
[122] A. Papoutsaki, P. Sangkloy, J. Laskey, N. Daskalova, J. Huang, and J. Hays. Webgazer: Scalable webcam eye tracking using user interactions. In *IJCAI*, 2016. (Cited on page 84.)
[123] P. Pobrotyn, T. Bartczak, M. Synowiec, R. Białobrzeski, and J. Bojar. Context-aware learning to rank with self-attention. *SIGIR eCom*, 2020. (Cited on pages 37 and 63.)
[124] R. Pradeep, R. Nogueira, and J. Lin. The expando-mono-duo design pattern for text ranking with pretrained sequence-to-sequence models. *arXiv preprint arXiv:2101.05667*, 2021. (Cited on page 28.)
[125] Prolific. Data Annotation. https://www.prolific.co/, 2023. (Cited on page 54.)
[126] Y. Qiao, C. Xiong, et al. Understanding the Behaviors of BERT in Ranking. *arXiv preprint arXiv:1904.07531*, 2019. (Cited on page 14.)
[127] T. Qin and T.-Y. Liu. Introducing letor 4.0 datasets. *arXiv preprint arXiv:1306.2597*, 2013. (Cited on page 62.)
[128] Qualtrics. XM. https://www.qualtrics.com/, 2023. (Cited on page 54.)
[129] S. Ramaswamy, R. Rastogi, and K. Shim. Efficient algorithms for mining outliers from large data sets. In *SIGMOD*, pages 427–438, 2000. (Cited on pages 68 and 75.)
[130] C. K. Reddy, L. Màrquez, F. Valero, N. Rao, H. Zaragoza, S. Bandyopadhyay, A. Biswas, A. Xing, and K. Subbian. Shopping queries dataset: A large-scale esci benchmark for improving product search. *arXiv preprint arXiv:2206.06588*, 2022. (Cited on page 101.)
[131] S. Robertson and H. Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009. (Cited on pages 1, 2, and 10.)

[132] A. Rosenbaum, S. Soltan, W. Hamza, M. Damonte, I. Groves, and A. Saffari. Clasp: Few-shot cross-lingual data augmentation for semantic parsing. In *AACL-IJCNLP*, pages 444–462, 2022. (Cited on page 102.)

[133] A. Rosenbaum, S. Soltan, W. Hamza, Y. Versley, and M. Boese. Linguist: Language model instruction tuning to generate annotated utterances for intent classification and slot tagging. In *Computational Linguistics*, pages 218–241, 2022. (Cited on page 102.)

[134] P. J. Rousseeuw and K. V. Driessen. A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41(3):212–223, 1999. (Cited on page 68.)

[135] P. Sapiezynski, W. Zeng, R. E Robertson, A. Mislove, and C. Wilson. Quantifying the impact of user attentionon fair group representation in ranked lists. In *WWW*, pages 553–562, 2019. (Cited on pages 3, 31, 48, and 72.)

[136] F. Sarvi. Understanding the effect of outlier items in e-commerce ranking. In *WSDM*, pages 1226–1227, 2023.

[137] F. Sarvi, N. Voskarides, L. Mooiman, S. Schelter, and M. de Rijke. A comparison of supervised learning to match methods for product search. *SIGIR Workshop on eCommerce*, 2020. (Cited on page 1.)

[138] F. Sarvi, M. Heuss, M. Aliannejadi, S. Schelter, and M. de Rijke. Understanding and mitigating the effect of outliers in fair ranking. In *WSDM*, pages 861–869, 2022. (Cited on pages 2, 3, 4, 52, 53, 55, 57, 65, 68, 69, 72, 76, 84, and 90.)

[139] F. Sarvi, M. Aliannejadi, S. Schelter, and M. de Rijke. How to make an outlier? Studying the effect of presentational features on the outlierness of items in product search results. In *CHIIR*, pages 346–350, 2023.

[140] F. Sarvi, A. Vardasbi, M. Aliannejadi, S. Schelter, and M. de Rijke. On the impact of outlier bias on user clicks. In *SIGIR*, pages 18–27, 2023. (Cited on pages 2 and 72.)

[141] F. Sarvi, M. Aliannejadi, S. Schelter, and M. de Rijke. Understanding visual saliency of outlier items in product search. *arXiv preprint arXiv:2503.23596*, 2025.

[142] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001. (Cited on pages 68 and 75.)

[143] H. Schütze, C. D. Manning, and P. Raghavan. *Introduction to information retrieval*, volume 39. Cambridge University Press Cambridge, 2008. (Cited on pages 1 and 31.)

[144] A. Schwager and C. Meyer. Understanding customer experience. *Harvard Business Review*, 85(2): 116, 2007. (Cited on page 82.)

[145] J. Shen, E. M. Reingold, and M. Pomplun. Guidance of eye movements during conjunctive visual search: The distractor-ratio effect. *Canadian Journal of Experimental Psychology/Revue Canadienne de Psychologie Expérimentale*, 57(2):76, 2003. (Cited on pages 73 and 76.)

[146] Y. Shen, X. He, J. Gao, L. Deng, and G. Mesnil. Learning semantic representations using convolutional neural networks for web search. In *WWW*, pages 373–374. ACM, 2014. (Cited on page 13.)

[147] S. W. Shi, M. Wedel, and R. Pieters. Information acquisition during online decision making: A model-based exploration using eye-tracking data. *Management Science*, 59(5):1009–1026, 2013. (Cited on page 82.)

[148] A. Singh and T. Joachims. Fairness of exposure in rankings. In *KDD*, pages 2219–2228, 2018. (Cited on pages 3, 31, 33, 34, 35, 41, 42, 43, 44, 48, 65, and 72.)

[149] A. Singh and T. Joachims. Policy learning for fairness in ranking. In *NeurIPS*, 2019. (Cited on pages 3, 31, 48, and 72.)

[150] P. Sondhi, M. Sharma, et al. A taxonomy of queries for e-commerce search. In *SIGIR*, pages 1245–1248. ACM, 2018. (Cited on pages 11 and 24.)

[151] J. Stoyanovich, K. Yang, and H. Jagadish. Online set selection with fairness and diversity constraints. In *EDBT*, pages 241–252, 2018. (Cited on page 47.)

[152] B. W. Tatler. The central fixation bias in scene viewing: Selecting an optimal viewing position independently of motor biases and image feature distributions. *Journal of Vision*, 7(14):4–4, 2007. (Cited on pages 74 and 87.)

[153] A. Treisman. Features and objects: The fourteenth Bartlett memorial lecture. *The Quarterly Journal of Experimental Psychology Section A*, 40(2):201–237, 1988. (Cited on page 76.)

[154] A. M. Treisman and G. Gelade. A feature-integration theory of attention. *Cognitive Psychology*, 12(1): 97–136, 1980. (Cited on pages 73, 74, 75, 76, 77, and 79.)

[155] A. Trotman, J. Degenhardt, and S. Kallumadi. The architecture of eBay search. In *eCom: The SIGIR 2017 Workshop on eCommerce*, 2017. (Cited on pages 11, 24, and 25.)

[156] M. Tsagkias, T. H. King, S. Kallumadi, V. Murdock, and M. de Rijke. Challenges and research opportunities in ecommerce search and recommendations. In *Sigir Forum*, volume 54, pages 1–23. ACM New York, NY, USA, 2021. (Cited on pages 1, 2, 9, and 10.)

[157] Z. Tupikovskaja-Omovie and D. Tyler. Eye tracking technology to audit google analytics: Analysing digital consumer shopping journey in fashion m-retail. *International Journal of Information Management*, 59:102294, 2021. (Cited on page 83.)

[158] C. Van Gysel, M. de Rijke, and E. Kanoulas. Learning latent vector spaces for product search. In *CIKM*, pages 165–174. ACM, October 2016. (Cited on pages 2 and 10.)

[159] D. Vandic, F. Frasincar, and U. Kaymak. Facet selection algorithms for web product search. In *CIKM*, pages 2327–2332. ACM, 2013. (Cited on page 11.)

[160] D. Vandic, S. Aanen, F. Frasincar, and U. Kaymak. Dynamic facet ordering for faceted product search engines. *TKDE*, 29(5):1004–1016, 2017. (Cited on page 11.)

[161] A. Vardasbi, H. Oosterhuis, and M. de Rijke. When inverse propensity scoring does not work: Affine corrections for unbiased learning to rank. In *CIKM*, pages 1475–1484, 2020. (Cited on pages 47, 62, 63, 68, and 72.)

[162] A. Vardasbi, M. de Rijke, and I. Markov. Mixture-based correction for position and trust bias in counterfactual learning to rank. In *CIKM*, pages 1869–1878, 2021. (Cited on page 62.)

[163] A. Vardasbi, F. Sarvi, and M. de Rijke. Probabilistic permutation graph search: Black-box optimization for fairness in ranking. In *SIGIR*, pages 715–725, 2022. (Cited on page 65.)

[164] R. Veale, Z. M. Hafed, and M. Yoshida. How is visual salience computed in the brain? insights from behaviour, neurobiology and modelling. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 372(1714):20160113, 2017. (Cited on page 75.)

[165] H. Wang, R. Langley, S. Kim, E. McCord-Snook, and H. Wang. Efficient exploration of gradient space for online learning to rank. In *SIGIR*, pages 145–154, 2018. (Cited on page 69.)

[166] H. Wang, M. J. Bah, and M. Hammad. Progress in outlier detection techniques: A survey. *Ieee Access*, 7:107964–108000, 2019. (Cited on pages 68, 75, and 76.)

[167] L. Wang and T. Joachims. User fairness, item fairness, and diversity for rankings in two-sided markets. In *SIGIR*, pages 23–41, 2021. (Cited on page 35.)

[168] Q. Wang, S. Yang, M. Liu, Z. Cao, and Q. Ma. An eye-tracking study of website complexity from cognitive load perspective. *Decision Support Systems*, 62:1–10, 2014. (Cited on page 83.)

[169] X. Wang, M. Bendersky, D. Metzler, and M. Najork. Learning to rank with selection bias in personal search. In *SIGIR*, pages 115–124, 2016. (Cited on pages 32 and 69.)

[170] X. Wang, N. Golbandi, M. Bendersky, D. Metzler, and M. Najork. Position bias estimation for unbiased learning to rank in personal search. In *WSDM*, pages 610–618, 2018. (Cited on pages 52, 60, 61, 63, and 69.)

[171] D. Weathers, S. Sharma, and S. L. Wood. Effects of online communication practices on consumer perceptions of performance uncertainty for search and experience goods. *Journal of Retailing*, 83(4): 393–401, 2007. (Cited on page 83.)

[172] K. Wisiecka, K. Krejtz, I. Krejtz, D. Sromek, A. Cellary, B. Lewandowska, and A. Duchowski. Comparison of webcam and remote eye tracking. In *2022 Symposium on Eye Tracking Research and Applications*, pages 1–7, 2022. (Cited on page 84.)

[173] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, and J. Brew. HuggingFace's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019. (Cited on page 17.)

[174] J. M. Wolfe. Guided search 2.0 a revised model of visual search. *Psychonomic bulletin & review*, 1: 202–238, 1994. (Cited on page 75.)

[175] J. M. Wolfe. What can 1 million trials tell us about visual search? *Psychological Science*, 9(1):33–39, 1998. (Cited on pages 73, 76, 77, and 78.)

[176] J. M. Wolfe. Visual search: How do we find what we are looking for? *Annual Review of Vision Science*, 6(1):539–562, 2020. (Cited on pages 75 and 81.)

[177] J. M. Wolfe, E. M. Palmer, and T. S. Horowitz. Reaction time distributions constrain models of visual search. *Vision research*, 50(14):1304–1311, 2010. (Cited on page 74.)

[178] C. Wu, M. Yan, and L. Si. Ensemble methods for personalized e-commerce search challenge at cikm cup 2016. *arXiv preprint arXiv:1708.04479*, 2017. (Cited on pages 10, 11, and 16.)

[179] X. Wu, H. Chen, J. Zhao, L. He, D. Yin, and Y. Chang. Unbiased learning to rank in feeds recommendation. In *WSDM*, pages 490–498, 2021. (Cited on pages 52, 69, and 72.)

[180] C. Xiong, Z. Dai, J. Callan, Z. Liu, and R. Power. End-to-end neural ad-hoc ranking with kernel pooling. In *SIGIR*, pages 55–64. ACM, 2017. (Cited on page 14.)

[181] J. Xu, X. He, and H. Li. Deep learning for matching in search and recommendation. In *SIGIR*, pages 1365–1368, 2018. (Cited on page 1.)

[182] H. Yadav, Z. Du, and T. Joachims. Policy-gradient training of fair and unbiased ranking functions. *arXiv preprint arXiv:1911.08054*, 2019. (Cited on pages 48 and 72.)

[183] K. Yang and J. Stoyanovich. Measuring fairness in ranked outputs. In *SSDBM*, pages 1–6, 2017. (Cited on page 47.)

[184] W. Yang, K. Lu, P. Yang, and J. Lin. Critically examining the "neural hype" weak baselines and the additivity of effectiveness gains from neural ranking models. In *SIGIR*, pages 1129–1132, 2019. (Cited on pages 12 and 24.)

[185] Y. Yue and T. Joachims. Interactively optimizing information retrieval systems as a dueling bandits problem. In *ICML*, pages 1201–1208, 2009. (Cited on page 69.)

[186] Y. Yue, R. Patel, and H. Roehrig. Beyond position bias: Examining result attractiveness as a source of presentation bias in clickthrough data. In *WWW*, pages 1011–1018, 2010. (Cited on pages 32, 47, 68, and 72.)

[187] M. Zehlike, F. Bonchi, C. Castillo, S. Hajian, M. Megahed, and R. Baeza-Yates. Fa*ir: A fair top-k ranking algorithm. In *CIKM*, pages 1569–1578, 2017. (Cited on page 47.)

[188] M. Zehlike, K. Yang, and J. Stoyanovich. Fairness in ranking: A survey. *arXiv preprint arXiv:2103.14000*, 2021. (Cited on page 47.)

[189] H. Zhang, T. Wang, et al. Improving semantic matching via multi-task learning in e-commerce. In *SIGIR Workshop on eCommerce*, 2019. (Cited on page 10.)

[190] Y. Zhang, D. Wang, and Y. Zhang. Neural IR meets graph embedding: A ranking model for product search. In *WWW*, pages 2390–2400. ACM, 2019. (Cited on pages 10, 12, and 16.)

[191] Y. Zhao, Z. Nasrullah, M. K. Hryniewicki, and Z. Li. LSCP: Locally selective combination in parallel outlier ensembles. In *ICDM*, pages 585–593, 2019. (Cited on page 68.)

[192] H. Zhuang, Z. Qin, R. Jagerman, K. Hui, J. Ma, J. Lu, J. Ni, X. Wang, and M. Bendersky. Rankt5: Fine-tuning t5 for text ranking with ranking losses. In *SIGIR*, pages 2308–2313, 2023. (Cited on pages 28 and 101.)

[193] S. Zhuang, H. Ren, L. Shou, J. Pei, M. Gong, G. Zuccon, and D. Jiang. Bridging the gap between indexing and retrieval for differentiable search index with query generation. *arXiv preprint arXiv:2206.10128*, 2022. (Cited on page 102.)

Summary

Ranking is at the core of information retrieval, from search engines to recommendation systems. The objective of a ranking model is to order items based on their degree of relevance to the user's information need, which is often expressed by a textual query. In product search, customers search through numerous options using brief, unstructured phrases, and the goal is to find not only relevant but also appealing products that match their preferences and lead to purchases. On the other side, there are the providers of the products who expect the ranking model to fairly expose their items to customers. These complications introduce unique characteristics that set product search apart from other types of search.

This thesis investigates the specific challenges of applying learning to rank models in product search and present methods to improve relevance, fairness, and effectiveness in this setting. We start by focusing on query-product matching based on textual data, as traditional information retrieval methods rely heavily on text to determine relevance. It has been shown that the vocabulary gap is larger in product search, mainly due to the limited and unstructured nature of queries and product descriptions. The vocabulary gap refers to the difference in the language used in queries and the terms found in product descriptions. In Chapter 2, we conduct a comprehensive evaluation of state-of-the-art supervised learning to match models, comparing their performance in product search. Our findings identify models that balance both accuracy and efficiency, offering practical insights for real-world applications.

Next, in Chapters 3 and 4 we address fairness in ranking on two-sided platforms, where the goal is to satisfy both groups of product search users at the same time. Accurate exposure estimation is crucial to achieve this balance. To this end, we introduce the phenomenon of outlierness in ranking as a factor that can influence the exposure-based fair ranking algorithms. Outlier items are products that deviate from others in a ranked list, due to distinct presentational features. We show empirically that these items attract more user attention and can impact exposure distribution in a list. To account for this effect, we propose OMIT, a method that reduces outlierness without compromising user utility or fairness towards providers. In the next chapter, we investigate whether outlier items influence user clicks. We introduce outlier bias as a new type of click bias, and propose OPBM. OPBM is an outlier-aware click model designed to account for both outlier and position bias. Our experiments show that in the worst case, OPBM performs similarly to the well-known Position-based model, making it a more reliable choice.

Finally, in Chapter 5 we explore how different presentational features influence user attention and perception of outliers in product search results. Through visual search and eye-tracking experiments, along with visual saliency modeling, we identify user scanning patterns and determine the role of bottom-up and top-down factors in guiding attention and shaping the perception of outliers.

Samenvatting

Rangschikken staat centraal in informatievoorziening, van zoekmachines tot aanbevelingssystemen. Het doel van een model om te rangschikken is om items te ordenen op basis van hun relevantie voor de informatiebehoefte van de gebruiker, die vaak wordt uitgedrukt in zoekopdrachten naar een tekstuele zoekvraag. Bij product doorzoeken klanten talloze opties met korte, ongestructureerde zinnen, waarbij het doel is om niet alleen relevante, maar ook aantrekkelijke producten te vinden die aansluiten bij hun voorkeuren en aankopen stimuleren. Aan de andere kant zijn er de aanbieders van producten, die verwachten dat het rangschikkingsmodel hun producten eerlijk aan klanten presenteert. Deze complicaties brengen unieke kenmerken met zich mee die productzoekopdrachten onderscheiden van andere vormen van zoekopdrachten.

Dit proefschrift onderzoekt de specifieke uitdagingen van het toepassen van *learning to rank* modellen bij productzoekopdrachten en presenteert methoden om relevantie, eerlijkheid en effectiviteit in deze context te verbeteren. We beginnen door ons te richten op het matchen van zoekvragen en producten op basis van tekstuele data, aangezien traditionele methoden voor informatievoorziening sterk afhankelijk zijn van tekst om relevantie te bepalen. Uit onderzoek blijkt dat de vocabulairekloof groter is bij productzoekopdrachten, voornamelijk door de beperkte en ongestructureerde aard van queries en productbeschrijvingen. De vocabulairekloof verwijst naar het verschil in taalgebruik tussen de zoeekvragen en de termen die voorkomen in productbeschrijvingen.

Vervolgens behandelen we in Hoofdstukken 3 en 4 eerlijkheid in ranking op tweezijdige platforms, waarbij het doel is beide groepen gebruikers van productzoekopdrachten tegelijkertijd tevreden te stellen. Een nauwkeurige schatting van *exposure* is cruciaal om dit evenwicht te bereiken. Daartoe introduceren we het fenomeen "outlierness" in ranking als een factor die invloed kan hebben op *exposure*-gebaseerde eerlijke ranking-algoritmen. *Outlier*-producten zijn producten die afwijken van andere producten in een gerangschikte lijst, door onderscheidende presentatiekenmerken. We tonen empirisch aan dat deze items meer gebruikersaandacht trekken en de *exposure*verdeling in een lijst kunnen beïnvloeden. Om dit effect te corrigeren, stellen we OMIT voor, een methode die *outlierness* vermindert zonder afbreuk te doen aan gebruiksgemak voor de gebruiker of eerlijkheid ten opzichte van aanbieders. In het volgende hoofdstuk onderzoeken we of *outlier*-items invloed hebben op gebruikerskliks. We introduceren *outlier*-bias als een nieuw type klikbias en stellen OPBM voor. OPBM is een *outlier*-bewust klikmodel dat rekening houdt met zowel *outlier*- als positie-bias. Onze experimenten tonen aan dat OPBM in het slechtste geval vergelijkbare prestaties levert als het bekende positie-gebaseerde model, waardoor het een betrouwbaardere keuze is.

Ten slotte onderzoeken we in Hoofdstuk 5 hoe verschillende presentatiekenmerken de aandacht van gebruikers en hun perceptie van *outliers* in productzoekresultaten beïnvloeden. Door middel van visueel zoeken, oogbewegingsonderzoek, en het modelleren van visuele opvallendheid identificeren we gebruikersscanpatronen en bepalen we de rol van *bottom-up* en *top-down* factoren bij het sturen van aandacht en vormen van de perceptie van *outliers*.