
Textual User Profiles for Search-based Recommendation

A dissertation submitted towards the degree
Doctor of Engineering (Dr.-Ing.)
of the Faculty of Mathematics and Computer Science
of Saarland University

by
Ghazaleh Haratinezhad Torbati

Saarbrücken
2025

Day of Colloquium

27.06.2025

Dean of the Faculty

Prof. Dr. Roland Speicher

Chair of the Committee

Prof. Dr. Anna Maria Feit

Reviewers

Prof. Dr. Gerhard Weikum

Prof. Dr. Andrew Yates

Prof. Dr. Jilles Vreeken

Academic Assistant

Dr. Soumi Das

Abstract

Personalization plays a central role in enhancing users’ online experiences by helping them navigate vast amounts of data across various applications and services. One realistic and understudied use case is *search-based recommendation*, which aims to integrate personalized recommendations with situational queries, reflecting users’ current intent. User profiling is a key component of personalized search and recommendations, and it is becoming increasingly important due to the growing availability of personal data generated through users’ online activities. Unlike user models based on clicks and interaction logs, textual profiles improve users’ understanding of the personalization process, thereby increasing trust and supporting transparency and scrutability. User-generated text—such as reviews, surveys, social media posts, and chats—offers nuanced insights into user interests and preferences, but it comes with complexity due to varied style and noisiness.

Utilizing alternative data sources for profiling in the context of search-based recommendation presents unique challenges, including: ensuring transparency and scrutability, addressing data sparsity and long-tail users, distilling valuable signals from noisy user-generated data, determining the extent of personalization achievable with different profiles, and overcoming the lack of publicly available data for search-based recommendation and alternative profiling sources. This dissertation addresses these challenges by the following contributions. The proposed methods are evaluated using a recommender system evaluation setup tailored to search-based recommendation, leveraging user studies and simulations where appropriate.

- *Sparse and Scrutable User Profiles.* We first make the case for using concise, questionnaire-based user profiles for personalization, which are captured from short questionnaires filled out by users. These profiles are an ideal starting point to investigate the viability of small profiles for personalization, as they are directly expressed by the end-user and contain minimal errors. Such profiles are inherently scrutable, given that their underlying source is a questionnaire. Our experiments, which include collecting user data in the form of questionnaires and judgments of query-item pairs, demonstrate the effectiveness of concise user profiles for search-based recommendation.
- *Chat-based User Profiles.* We propose tapping into a novel source for user profiling—user-to-user conversations—and leveraging such implicit signals for

personalization. While chats as user profiles are less interpretable than questionnaires, they are still more comprehensible than non-textual sources, such as user click logs. We study the extent of personalization performance by comparing concise questionnaire-based profiles with richer chat-based profiles for the same users. Our extensive multi-stage user study, which collects chat data in addition to questionnaires and assessments, shows the improved performance of personalization results using both profiling sources over non-personalized results.

- *Concise User Profiles from Review Texts.* We introduce an approach for constructing concise profiles from long and noisy review texts, combining the best of the previously researched settings by automatically constructing user profiles. This eliminates the need for users to manually fill out profiles while ensuring that the profiles remain concise, allowing users to understand and scrutinize them. We propose several methods for profile construction, including extracting informative cues using language statistics and generating profiles using large language models. Additionally, we propose soft-labeling techniques to address the lack of explicitly labeled negative data for training the recommender model. Extensive experiments compare several user profiling approaches and demonstrate that selecting informative text snippets and filtering out noise leads to better personalization performance.

Kurzfassung

Personalisierung spielt eine zentrale Rolle bei der Verbesserung digitaler Nutzererlebnisse, indem sie dabei hilft, sich in der Vielzahl an Informationen innerhalb verschiedenster Anwendungen und Dienste zurechtzufinden. Ein realistisches und bislang wenig erforschtes Szenario stellt die *suchbasierte Empfehlung* dar. Ziel ist es, personalisierte Empfehlungen mit situativen Suchanfragen zu verknüpfen und dadurch die aktuelle Nutzerintention abzubilden. Die Erstellung aussagekräftiger Nutzerprofile bildet dabei ein zentrales Element personalisierter Such- und Empfehlungsdienste – insbesondere angesichts der zunehmenden Verfügbarkeit persönlicher Daten, die durch Online-Aktivitäten generiert werden. Im Gegensatz zu Interaktionsdaten wie Klicks und Protokollen bieten textuelle Nutzerprofile ein tieferes Verständnis des Personalisierungsprozesses, stärken das Vertrauen der Nutzenden und fördern Transparenz sowie Nachvollziehbarkeit. Nutzergenerierte Texte – beispielsweise Rezensionen, Umfrageantworten, Social-Media-Beiträge oder Chats – ermöglichen differenzierte Einblicke in Interessen und Präferenzen, stellen jedoch aufgrund ihres variierenden Stils und hoher Rauschanteile auch besondere Herausforderungen dar.

Die Nutzung solcher alternativer Datenquellen zur Profilerstellung, im Kontext suchbasierter Empfehlungen, wirft eine Reihe spezifischer Herausforderungen auf: die Sicherstellung von Transparenz und Nachvollziehbarkeit, der Umgang mit Datenknappheit und Long-Tail-Nutzenden, die Extraktion relevanter Signale aus verrauschten Textdaten, die Abschätzung des erreichbaren Personalisierungsgrads unterschiedlicher Profiltypen sowie der Mangel an öffentlich verfügbaren Daten für diesen speziellen Anwendungsfall. Diese Dissertation adressiert diese Herausforderungen mit folgenden Beiträgen. Unsere Ansätze werden im Rahmen eines speziell für suchbasierte Empfehlungen konzipierten Evaluations-Setups für Empfehlungsdienste validiert – unter Einbeziehung von Nutzerstudien und Simulationen, wann immer sinnvoll.

- *Spärliche und nachvollziehbare Nutzerprofile.* Wir argumentieren für den Einsatz kompakter, fragebogenbasierter Profile, die direkt von den Nutzenden über kurze Fragebögen bereitgestellt werden. Diese Profile eignen sich ideal, um das Potenzial kleiner, aber ausdrucksstarker Profile für die Personalisierung zu untersuchen, da sie direkt vom Endnutzenden stammen und kaum fehlerbehaftet sind. Ihre Herkunft aus standardisierten Fragebögen macht sie zudem von Natur aus nachvollziehbar. Unsere Experimente – einschließlich der Erhebung von

Fragebogendaten sowie der Bewertung von Suche-Empfehlung-Paaren – belegen die Effektivität dieser Profilform im Kontext suchbasierter Empfehlungen.

- *Chat-basierte Nutzerprofile.* Wir schlagen vor, Nutzer-zu-Nutzer-Gespräche als neuartige Quelle impliziter Signale für die Profilerstellung zu erschließen. Auch wenn Chatverläufe weniger unmittelbar interpretierbar sind als Fragebögen, bieten sie dennoch mehr Transparenz als nicht-textuelle Quellen wie Klickprotokolle. Wir analysieren das Personalisierungspotenzial, indem wir kompakte, fragebogenbasierte Profile mit umfangreicheren, chatbasierten Profilen derselben Nutzenden vergleichen. Eine umfassende, mehrstufige Nutzerstudie, in der sowohl Chatdaten als auch Fragebögen und Bewertungen erhoben werden, zeigt eine deutliche Steigerung der Personalisierungsleistung bei Kombination beider Profilierungsquellen gegenüber nicht-personalisierten Empfehlungen.
- *Kompakte Nutzerprofile aus Rezensionstexten.* Wir präsentieren einen Ansatz zur automatisierten Erstellung kompakter Nutzerprofile aus langen und veräuschten Rezensionstexten. Dieser Ansatz vereint die Vorteile der zuvor untersuchten Szenarien und ermöglicht eine Profilerstellung ohne aktiven Nutzereingriff – bei gleichzeitig hoher Nachvollziehbarkeit. Wir entwickeln mehrere Verfahren zur Profilerstellung, darunter sprachstatistische Methoden zur Merkmalsextraktion sowie generative Ansätze unter Einsatz großer Sprachmodelle. Ergänzend führen wir Soft-Labeling-Techniken ein, um das Fehlen explizit negativer Trainingsbeispiele im Empfehlungsdienst zu kompensieren. Unsere umfassenden Experimente zeigen, dass die gezielte Auswahl informativer Textsegmente und das Herausfiltern irrelevanter Inhalte die Personalisierungsqualität deutlich verbessern.

Acknowledgments

I am grateful to Gerhard Weikum, my supervisor, for his support and guidance throughout this journey. This thesis would not have been possible without his insights, encouragement, and steady mentorship.

I would also like to thank my collaborators, Andrew Yates and Anna Tigunova, for many constructive discussions and their valuable input throughout the years.

Warm thanks to Prof. Jilles Vreeken for reviewing my work and providing insightful feedback, and to Prof. Anna Maria Feit and Dr. Soumi Das for serving on my thesis committee and supporting the defense process.

Thanks to all of my colleagues and friends in D5, who created a fun, motivating, supportive, and intellectually stimulating environment to work and grow in.

I am deeply grateful to my friends and family — especially my parents, Fatemeh and Alireza — who have always shown me unconditional love and supported every decision I've made.

Last but not least, my heartfelt thanks to my partner, Amir, for standing by me through the ups and downs and always encouraging me to be my best self.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Research Challenges	3
1.3	State of the Art and its Limitations	4
1.4	Contributions	6
1.5	Publications	8
1.6	Organization	9
2	Background	11
2.1	Recommendation Systems	12
2.1.1	Task Definition	12
2.1.2	Recommender Models	14
2.1.3	Challenges	18
2.2	Personalized Search	19
2.2.1	General Concepts of Information Retrieval	20
2.2.2	Personalized Information Retrieval	25
2.2.3	Personalized Entity Search	27
2.3	User Profiling	28
2.4	Evaluation	29
2.4.1	Metrics	30
2.4.2	Significance Testing	31
3	Sparse and Scrutable User Profiles	33
3.1	Introduction	34
3.2	Methodology	36
3.2.1	Search-based Recommendation	37
3.2.2	User Profiling	37
3.2.3	Re-ranking Methods	38
3.3	Data Collection	40
3.3.1	Queries and Documents	40
3.3.2	User Study	41
3.4	Experiments	41
3.4.1	Research Questions	41

3.4.2	Experimental Setup	42
3.5	Results	43
3.5.1	Main Findings	43
3.5.2	Ablation Study	44
3.6	Related Work	45
3.7	Conclusion	46
4	Chat-based User Profiles	47
4.1	Introduction	48
4.2	Methodology	50
4.2.1	Overview	50
4.2.2	Entity Expansion	51
4.2.3	Domain-specific Vocabulary Weighting	53
4.2.4	Re-ranking Methods	53
4.3	Data Collection	55
4.3.1	Domains, Queries, and Documents	55
4.3.2	User Study	56
4.4	Experiments	58
4.4.1	Research Questions	58
4.4.2	Experimental Setup	59
4.5	Results	61
4.5.1	Main Findings: RQ0 and RQ1	62
4.5.2	Domain Vocabularies: RQ2	63
4.5.3	Entity Expansion: RQ3	64
4.6	Related Work	64
4.7	Conclusion	66
5	Concise User Profiles from Review Texts	67
5.1	Introduction	68
5.2	Methodology	70
5.2.1	User Profile Construction Approaches	71
5.2.2	Ranking Model	75
5.2.3	Negative Training Samples from Unlabeled Data	77
5.3	Experiments	79
5.3.1	Rationale	79
5.3.2	Datasets	79
5.3.3	Experimental Setup	81
5.4	Results	83
5.4.1	Comparison of CUP against Baselines	83
5.4.2	Comparison of CUP Configurations	86
5.4.3	Efficiency of CUP	88
5.4.4	Influence of Interaction Density	89
5.5	Analysis of User Profiles	91
5.6	Related Work	92

5.7	Conclusion	94
6	SIRUP System	99
6.1	Architecture and Implementation	100
6.1.1	Profile Construction	100
6.1.2	Candidate Filtering	100
6.1.3	Personalized Ranking	101
6.1.4	Data	101
6.2	SIRUP Platform	101
6.2.1	Constructing User Profiles	102
6.2.2	Defining Search Contexts	105
6.2.3	Recommendation Results	108
6.3	Related Work	108
6.4	Conclusion	109
7	Conclusion	111
7.1	Summary	111
7.2	Outlook	112
	List of Figures	116
	List of Tables	118
	Bibliography	118

Chapter 1

Introduction

Contents

1.1	Motivation	1
1.2	Research Challenges	3
1.3	State of the Art and its Limitations	4
1.4	Contributions	6
1.5	Publications	8
1.6	Organization	9

1.1 Motivation

Recommendation systems have become an integral part of users' online experience, with ever-growing applications, including entertainment (e.g., movies, books, and music), e-commerce, travel, news, financial advice, and healthcare. The information overload, combined with users' limited time, attention, and resources, places the content presented to them at high stakes. In addition to the quality of the provided content, users are seeking to understand and have control over the recommendation process.

Increasing usage of online services and applications has led to vast amounts of user data in various forms, such as user *interactions* (e.g., clicks, likes, ratings, reviews, purchases) with online services (e.g., e-commerce websites, video streaming, music repositories, book communities), general web browsing, social media activities, and chats. Such complex, noisy, and heterogeneous data requires judicious data selection for the best recommendation performance.

One of the prominent signals for recommenders is the *collaborative* signal encoded in the *user-item interaction matrix*, relying on the assumption that users who liked the

same items in the past will have similar interest in the future. These models require abundant amount of interaction data to effectively capture the collaborative signal [Su and Khoshgoftaar, 2009]. They underperform for long-tail users and items, and they lack interpretability toward humans.

In contrast, in settings with sparse data and long-tail users and items, *content-based* approaches are superior. Classical content-based approaches utilize item metadata, categorical and textual, to create both item and user profiles. The main shortcoming of such user profiling is that it creates the same profile for two users who liked the same item without capturing the individual underlying reasons. On the other hand, user-written text (e.g., reviews, posts, chats) as a source of data has been less studied, due to its complex and challenging form. Such data sources, however, are the most faithful to the user interest and style, and contain novel information that cannot be found in user-item interactions and item metadata. Example excerpts of user-written text are shown in Figure 1.1.



Figure 1.1: Excerpts of user-written text.

Typically, recommendations are solely made using the users' profiles or historical interactions without any additional input, such as by populating users' homepages or providing a stream of content. This paradigm fails to take into account the current user intent and context at the time of interacting with the system, leading to the emergence of *context-aware* recommendations [Adomavicius et al., 2022, Jannach and Zanker, 2024]. This thesis focuses on a special type of context, directly and explicitly provided by the user in form of a *textual query*, addressing the *search-based recommendation* paradigm. This paradigm also differs from *information retrieval*

where the objective is to retrieve documents that are relevant to the query without considering users’ specific interests. For example, consider two users with different interests: Alice loves vampire stories, and Bob enjoys psychological thrillers. They are both searching for a book with the theme of *weird characters with a plot twist*, as depicted in Figure 1.2. They would benefit from the search-based recommender system by being able to explicitly verbalize their intent as a textual query, and by the system utilizing their interests and reading history when suggesting “Darkness Before Dawn” to Alice and “Secrets of the Weird” to Bob.



Figure 1.2: Search-based recommendation, for two users with different backgrounds for the same input query.

Problem Statement. In this thesis we focus on textual user profiles as a way for transparent and scrutable recommendation, such that users can easily understand, edit, and extend their profiles. Our focus on user-generated text as the source for user profiling calls for methods to extract informative cues from noisy and complex text. Search-based recommendation is a paradigm where both users’ long-term profiles and their current query intent are utilized to guide the recommendation process. This paradigm is an underexplored research area, and the lack of public data with both user preferences and queries makes it more challenging to study. This thesis explores different kinds of textual user profiles, namely questionnaire-based, chat-based, and review-based, to enable personalization in the difficult case of search-based recommendation.

1.2 Research Challenges

We identify the following challenges while investigating alternative textual sources for user profiling in the search-based recommendation paradigm.

C1: Transparency and Scrutability of Recommendation. Textual user profiles improve the transparency of the recommendation process, as they are more human-comprehensible compared to latent signals. Several factors affect the users’ understanding of their profiles. For instance, a lengthy user profile aggregated from all user data can reduce user control and makes it difficult to elicit useful information effectively. Other factors include the readability and linguistic style of the final profile.

C2: Handling Long-tail Users. Users with sparse interaction data, niche interests, or highly diverse tastes often suffer from lower-quality recommendation services. The literature provides limited coverage of long-tail and cold-data users. It is important to consider these user types, when designing and evaluating personalization models.

C3: Eliciting Informative Cues from Noisy User Data. User-generated text is noisy and complex, often containing a mixture of elements such as:

- Background information (e.g., “I’m a retired teacher”),
- General emotions and statements (e.g., “I’m in a good mood today”),
- Item properties (e.g., “The story depicted the postwar Berlin”, “This location gave easy access to both scuba diving and hiking”),
- Sentiment and emotional expressions toward an item of interest (e.g., “What a great story”, “Brings back lovely memories”), and
- General interests (e.g., “I enjoy comedy”, “I’m a nature lover”).

Only a subset of this information contributes to improving recommendation performance, while other aspects may add noise and reduce personalization quality.

C4: Extent of Personalization with Various Profiles. It is important to study the impact of profile sources and profiling techniques on personalization quality. Data sources such as chats and questionnaires are barely explored in this context.

C5: Lack of Public Data for Search-based Recommendation Scenario. Public recommendation datasets only contain user-item interactions, sometimes accompanied by item metadata and user reviews. However, they often lack any representation of user intent, such as a textual query, making it challenging to investigate this paradigm effectively.

1.3 State of the Art and its Limitations

Collaborative filtering recommenders require sufficient overlapping user-item interaction data to perform effectively. The higher the connectivity and the denser the data, the better the collaborative signals that can be captured. In scenarios with extremely sparse interaction data and long-tail users or items, leveraging item metadata and user profiles becomes essential, naturally leading to the adoption of content-based approaches [Ricci et al., 2022].

The majority of content-based recommenders rely on item metadata (e.g., title, categories, description) to create user representations. The core idea behind these approaches is to match items with similar metadata to those in the user’s history (i.e., the user profile). However, this strategy is often insufficient, as it fails to account for the unique traits of individual users and the specific reasons they liked a particular item. Likewise, demographic-based user profiles (e.g., age, nationality, gender) are too simplistic and fail to capture users’ personal interests and contextual factors.

User reviews have been studied as an instance of user-generated text, primarily to create both user and item representations. [Zheng et al., 2017] concatenate all reviews to create the representations, while [Chen et al., 2018, Pugoy and Kao, 2020] apply attention mechanisms (review-level and word-level respectively) for a weighted pooling of useful review and tokens. [Pugoy and Kao, 2021] propose selecting sentences from reviews by clustering similar sentences and choosing a representative from each cluster; they consider a relatively large number of clusters in their experiments (40% of total sentences). However, none of these methods create explicit and concise textual user profiles, thus limiting the users’ ability to fully understand and control their profiles.

A recent trend is to construct a textual user profile for transparent and scrutable recommendation [Radlinski et al., 2022]. Tag-based user profiles are among the first approaches to address these problems [Balog et al., 2019, Mysore et al., 2023]. Closer to our research are methods using a natural language user profiles, which are richer and less restrictive than tag-based ones. For example, [Sanner et al., 2023] use user-written profiles and measure the ranking performance of a small item set in a user study setting, while [Ramos et al., 2024] utilize user reviews as a source to generate user profiles and investigate rating prediction task using the generated profiles. However, these studies do not address the difficult task of item ranking in a larger item corpus, and provide limited exploration of user profiling techniques.

Personalized information retrieval is another field of research focused on tailoring search results to the individual user taste, though it primarily addresses general information needs [Ghorab et al., 2013]. One of the closest scenarios to search-based recommendation is personalized product search. For example, [Ai et al., 2017, Zhang et al., 2020] address this problem by jointly learning representations for items, users and queries. A drawback of these methods is their use of synthetic or proprietary datasets. Furthermore, recent works optimize a single model for both search and recommendation [Zamani and Croft, 2020, Penha et al., 2024], demonstrating benefits over using two separate models. However, they do not explore the search-based recommendation paradigm, where both the query and user profile are simultaneously used as inputs to the model, as opposed to being considered two separate tasks. Overall, the search-based recommendation problem remains underexplored, and the lack of public dataset containing both user interactions and their queries creates a major bottleneck.

1.4 Contributions

In this thesis, we address the challenges in exploring textual user profiles for search-based recommendation and propose the following solutions:

Transparent and Scrutable User Profiles. An overarching theme of this work is ensuring that user profiles are scrutable and understandable by employing explicit text-based user profiles rather than latent ones. More specifically, in Chapter 3 and Chapter 5, we address this challenge by maintaining concise and minimal user profiles. This allows users to review their profiles in a short amount of time and easily edit them to reflect their preferences. In Chapter 3, we propose using user-filled questionnaires as a source for personalization. This approach extends beyond simple demographics by capturing users’ in-domain interests, an area that has been underexplored in prior works. In Chapter 5, we use user-written reviews as a source for user profiling, imposing constraints on the size and style of the user profile to ensure it is easily readable and comprehensible by humans. To make the best use of the limited profile budget, we propose several profiling techniques to extract informative signals from reviews and filtering out noisy ones. To the best of our knowledge, works on creating concise user profiles from user-generated text is an emerging and underexplored area of research. Examples of questionnaire-based and review-derived user profiles are shown in Figure 1.3.

Sparse Data Setting and Long-tail Users and Items. The experimental setups in this thesis focus on extremely sparse data with long-tail users and items, in contrast to the data-rich setups commonly employed in prior works, which mostly rely on collaborative signals. In Chapters 3 and 4, the experimental data is collected through user studies, which inherently results in sparse datasets. In Chapter 5, we intentionally avoid common data preprocessing practices that create dense version of datasets and remove the majority of long-tail users and items. Instead, we focus on text-rich but interaction-sparse setup. Additionally, going beyond aggregated evaluation metrics, we group users and items based on their interaction density and report performance for each group. This allows for a deeper understanding of how different methods perform across different populations and groups.

Eliciting Informative Cues from Noisy User Reviews. User-generated reviews are a rich source of text, but they contain both irrelevant content and valuable information. Our primary interest lies in capturing *why* a user liked the item they reviewed, while filtering out generic sentiments or irrelevant details. In Chapter 5, we address this challenge by proposing techniques to either extract or generate the most informative signals, leveraging both linguistic methods and generative models.

Extent of Personalization with Various Profiles. A significant challenge addressed in this thesis is evaluating the extent of personalization performance using different types of user profiles. We investigate two novel sources of profiling for

Travel	
Which countries have you recently traveled to?	Switzerland, Netherland, Germany, Georgia, Italy
Which locations/activities did you enjoy the most?	Hiking in the swiss alps
Name 3 places that you would like to visit?	Egypt, San Francisco, Tokyo

Book	
Please name 3 of your favorite books. And explain why you love them in a sentence.	<p>Harry potter because I got lost in it's magic world and I'm still believing in it.</p> <p>Ikigai it gave me courage to start thinking about what I'm living for.</p> <p>Factfulness it has changed my worldview which is based on facts now.</p>
Which Genres of books do you like?	science fiction and related to sport science

Food	
What are your favorite meals/foods?	Chicken Masala: I like it because it is so spicy
What is your regular cuisine?	Rice and kinds of stews.
Which world cuisines?	<input checked="" type="checkbox"/> Indian, Italian, Mexican, Middle Eastern

Abstractive Profile
I enjoy reading horror, dark fiction, and suspenseful stories. I appreciate authors who can craft a unique narrative and well - developed characters. I'm drawn to books that explore the human condition, psychology, and the supernatural.

Keyword Profile
Horror, dark fiction, Halloween, anthology, werewolf, supernatural, fantasy, thriller, suspense, sci-fi, paranormal, apocalyptic, post-apocalyptic, zombie, virus, pandemic, contamination, outbreak, monster, creature

Sentence Profile
"If Jeff Strand and a seedy motel room had a lovechild, that would be Adam.", "Tijuana Donkey Showdown is the long-awaited sequel to Adam Howe's story Damn Dirty Apes, from his stellar novella collection Die Dog or Eat the Hatchet.", "Wrathbone: Beautiful, haunting, horrifying."

(a) Questionnaire-based User Profile.

(b) Review-based User Profiles.

Figure 1.3: Examples of User Profiles.

personalization: questionnaires in Chapter 3 and chats in Chapter 4. To facilitate this investigation, we conduct extensive data collection, including a longitudinal user study detailed in Chapter 4. In this study, we collect both types of user data, questionnaires and chats, from the same users, along with their item assessments. This setup allows for a direct comparison of the performance of sparse questionnaire-based profiles against richer chat-based profiles for the same users. Furthermore, in Chapter 5, we examine the extent of personalization with different user profiles created from the same source using different profiling techniques.

Data Collection and Simulation for Search-based Recommendation. The lack of publicly available datasets containing both user-item interactions and queries poses a significant challenge for studying search-based recommendation. In Chapters 3 and 4, we address this by simulating users interacting with such system. Specifically, we acquire queries of interest from users and collect their assessments on multiple items for each query, with these items retrieved from a commercial search engine based on the query. In Chapter 5, using public datasets with lack of true user queries, we simulate a search-based scenario during evaluation. To achieve this, we create a pool

of items that are textually similar to the positive test item, under the assumption that these items would be retrieved using the same query in a basic text-matching retrieval system. Additionally, in Chapter 6, we present our interactive system, which allows users to explore personalization within the search-based recommendation paradigm by adding their own textual queries or utilizing the query-by-example functionality.

1.5 Publications

The research presented in this thesis includes material that has been published in the following papers, with the author of this dissertation being the main author.

Chapter 3 (Sparse and Scrutable User Profiles) is based on:

- [Ghazaleh Haratinezhad Torbati](#), Andrew Yates, and Gerhard Weikum. (2020, March). **Personalized Entity Search by Sparse and Scrutable User Profiles**. In *CHIIR'20: The Conference on Human Information Interaction and Retrieval*. (pp. 427–431).

Chapter 4 (Chat-based User Profiles) is based on:

- [Ghazaleh Haratinezhad Torbati](#), Andrew Yates, and Gerhard Weikum. (2021, March). **You Get What You Chat: Using Conversations to Personalize Search-Based Recommendations**. In *ECIR'21: The 43th European Conference on Information Retrieval*. (pp. 207–223).

Chapter 5 (Concise User Profiles from Review Texts) is based on:

- [Ghazaleh Haratinezhad Torbati](#), Anna Tiginova, Andrew Yates, and Gerhard Weikum. (2025, April). **CUP: a Framework for Resource-Efficient Review-Based Recommenders**. In *ECIR'25: The 47th European Conference on Information Retrieval*. (pp. 360 - 375).
- [Ghazaleh Haratinezhad Torbati](#), Gerhard Weikum, and Andrew Yates. (2023, April). **Search-based Recommendation: the Case for Difficult Predictions**. In *WWW'23: The ACM Web Conference*. (pp. 318–321).
- [Ghazaleh Haratinezhad Torbati](#), Anna Tiginova, and Gerhard Weikum. (2023, September). **Unveiling challenging cases in text-based recommender systems**. In *PERSPECTIVES'23: The 3rd Workshop on Perspectives on the Evaluation of Recommender Systems, co-located at at ACM Recommender Systems Conference*.

Chapter 6 (SIRUP System) is based on:

- Ghazaleh Haratinezhad Torbati, Anna Tigunova, and Gerhard Weikum. (2024, March). **SIRUP: Search-based Book Recommendation Playground**. In *WSDM'24: The 17th ACM International Conference on Web Search and Data Mining* (pp. 1062–1065).

The following publications were completed during this period but are not included in this thesis:

- Anna Tigunova, Ghazaleh Haratinezhad Torbati, Andrew Yates, and Gerhard Weikum. (2024, October). **STAR: Sparse Text Approach for Recommendation**. In *CIKM'24: The 33rd ACM International Conference on Information and Knowledge Management*. (pp. 4086–4090).
- Lukas Lange, Marc Müller, Ghazaleh Haratinezhad Torbati, Dragan Milchevski, Patrick Grau, Subhash Chandra Pujari, and Annemarie Friedrich. (2024, May). **AnnoCTR: A Dataset for Detecting and Linking Entities, Tactics, and Techniques in Cyber Threat Reports**. In *LREC-COLING'24: The Joint International Conference on Computational Linguistics, Language Resources and Evaluation* (pp. 1147–1160).

1.6 Organization

The rest of this thesis is organized as follows. In Chapter 2, we discuss the necessary background, focusing on recommendation systems, personalized search, and user profiling. Chapter 3 explores sparse questionnaire-based user profiles. In Chapter 4, we investigate chat-based user profiles and conduct an extensive user study to compare them with questionnaire-based profiles. Chapter 5 presents techniques for creating concise user profiles from review texts. Chapter 6 introduces our system for search-based recommendation using review-based profiling methods. Finally, in Chapter 7, we conclude the thesis and propose potential future research directions.

Chapter 2

Background

Contents

2.1	Recommendation Systems	12
2.1.1	Task Definition	12
2.1.2	Recommender Models	14
2.1.3	Challenges	18
2.2	Personalized Search	19
2.2.1	General Concepts of Information Retrieval	20
2.2.2	Personalized Information Retrieval	25
2.2.3	Personalized Entity Search	27
2.3	User Profiling	28
2.4	Evaluation	29
2.4.1	Metrics	30
2.4.2	Significance Testing	31

In this chapter, we present the necessary background for our work on **search-based recommendation**. Recommender system (RS) and Personalized Information Retrieval (PIR) are closely associated, sharing the same goal of presenting the most relevant and interesting information to the user and assisting the user in overcoming information overload. There are two main differences between the two paradigms: The first is that the PIR system addresses the user's information need in response to an explicit query submitted by the user, while RS filters a given data repository or a continuous stream of information without a specific information request from the user. The second difference is in the domain of information: work on RS mostly focuses on retail and entertainment domains, while PIR is mainly concerned with general information needs. The practical focus of this thesis lies between the two worlds, recommending items in the domain of entertainment or retail, but with a

specific query submitted by the user. The most similar research direction to ours is *personalized entity search*, where the entity domain is often retail products.

Sections 2.1 and 2.2 elaborate on the two key research areas of personalization, focusing on recommender systems [Ricci et al., 2015] and personalized search [Ghorab et al., 2013], respectively. In Section 2.3, we examine user profiling and modeling in the literature, including the use of textual features. Finally, Section 2.4 describes the standard evaluation practices in recommender systems and personalized IR.

2.1 Recommendation Systems

Recommender systems gained prominence in the 1990s and have remained relevant in both academic and industrial research due to their essential role in helping users navigate information overload and simplifying the decision-making process [Adomavicius and Tuzhilin, 2005]. The number of applications and domains benefiting from recommender systems continues to grow, ranging from well-established areas like streaming services (video and audio), books, news, and e-commerce, to emerging fields such as health and finance.

Recommender system methods encompass a broad area of research, including *interaction-based* methods like matrix factorization and auto-encoders, and *content-based* approaches that leverage item features, including tags and descriptions. A key concept in recommender systems is *user-item interactions*, which captures users' actions with respect to items. These actions can take the form of *explicit feedback*, such as ratings, likes, and dislikes, or *implicit feedback*, inferred from logged behavior such as clicks, add-to-list actions, and content consumptions (e.g., listening, watching, reading, purchasing), each with varying levels of confidence [Hu et al., 2008].

In this section, we classify recommendation tasks in Subsection 2.1.1, explore different recommender models in Subsection 2.1.2, and discuss challenges in recommender systems in Subsection 2.1.3.

2.1.1 Task Definition

The recommendation problem can be broadly categorized into two primary models: rating prediction and ranking [Aggarwal, 2016].

- *Rating Prediction*: This task aims to estimate the rating a user would give to an unseen item, typically represented as a natural number from 1 to 5, reflecting the user's likely satisfaction with the item.
- *Ranking*: Also known as the *top-k recommendation problem*, the goal of this task is to provide a ranked list of items, with the most relevant or interesting items for the user placed at the top. In practice, users are generally more interested in seeing a shortlist of potentially interesting items rather than the

estimated ratings for individual items, making this formulation more aligned with the application of recommender systems [Adamopoulos, 2013].

Recommender system research can also be categorized with respect to different considerations of modeling the user-item interactions.

- *Direct recommendation*, also noted as *Traditional* recommender system in some literature, takes a set of user interactions as input and predicts the user’s likely future interactions. It models the user’s long-term preferences and creates a static user model under the assumption that the entire user interaction history is equally important without considering and modeling the temporal or sequential aspects of the interactions. Many prominent recommender models such as matrix factorization fall into this category.
- *Sequential recommendation* takes a sequence of user interactions as input and predicts the user’s next interaction by modeling the complex sequential dependencies, based on the premise that the order of interactions is as important as the interaction history. It captures both the long-term and short-term user preferences, mainly emphasizing the more recent ones. Sequential recommendation has gained popularity in recent years due to its application in short-attention content such as personalized content feed and music playlists and video streaming services [Wang et al., 2019, Fang et al., 2020].
- *Session-based recommendation* is a specific type of sequential recommendation that focuses on recommendations within one session. This type of recommender only models the user’s short-term interest in the ongoing session [Wang et al., 2022]. Conversational recommendation is another line of research where the emphasis is on the ongoing real-time chat session where the system engages the user via dialogue to refine the recommendation [Jannach et al., 2022].
- *Context-aware recommendation* goes beyond just considering user interest and history; they also take into account specific contextual factors or situations, to find the most relevant items [Adomavicius et al., 2022, Baltrunas and Ricci, 2014]. Such systems integrate long-term user history with session-specific contextual information, which may be explicitly provided or inferred from implicit signals [Ding et al., 2019]. Examples of context include temporal aspects (e.g., time of day, day of the week), spatial data (e.g., location), psychological factors (e.g., mood, emotions), social settings (e.g., companionship), and users’ search intents, such as a textual query.

In this thesis, we focus on the **ranking task** within the recommendation problem, as it offers a more realistic formulation of user needs. Our approach to modeling user preferences is aligned with the **direct recommendation method**, which does not consider the sequential order of user interactions history. This formulation is particularly suitable for domains requiring long attention spans, such as books and travel destinations. We explore a special case of context-aware recommendation,

search-based recommendation, where the contextual information is users' search intent, such as a textual query or a query-by-example.

2.1.2 Recommender Models

There are two main classes of recommender models: *interaction-based* (also known as *collaborative filtering*) and *content-based*. The former relies on user-item interactions, while the latter primarily utilizes users' and items' attribute information. In addition to these primary methods, there are also *knowledge-based* recommenders, which address explicit user requirements, and *hybrid* recommender systems, which combine the advantages of various models. In this section, we focus on these types of recommenders while excluding others that are outside the scope of this thesis, such as *knowledge-graph-based* recommenders [Wu et al., 2023], which use the structure and the relationships between entities in a knowledge graph to provide recommendations.

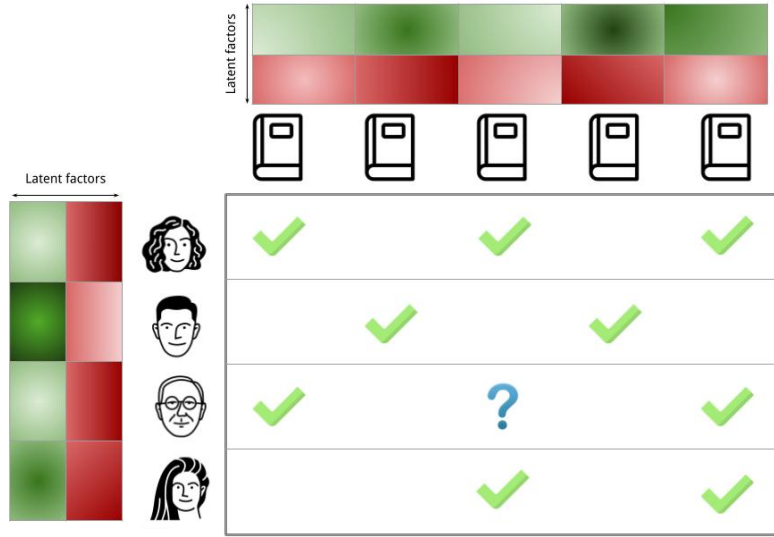


Figure 2.1: An illustration of collaborative filtering (CF) showing the user-item interaction matrix and learned latent representations for users and items. The figure demonstrates that similar users and items are represented more closely in the latent space, enabling recommendation. For example, the third user is likely to be interested in the item in question (“?”) because similar users have liked it.

Collaborative filtering-based recommenders (CF)

Collaborative filtering is one of the most successful and widely adopted techniques in practice, based on the fundamental assumption that like-minded users, who have shared similar interests in the past, are likely to share similar interests in the future (Figure 2.1) [Ricci et al., 2015, Adomavicius and Tuzhilin, 2005]. CF methods can be divided into two main categories: *neighborhood-based* (also known as *memory-based*) and *model-based* [Aggarwal, 2016].

Neighborhood-based CF calculates the similarities between users and items by directly utilizing the stored user-item interaction data. This can be done in one of two ways: *user-based* or *item-based*. In user-based CF [Konstan et al., 1997, Tan and He, 2017], the utility of an item to a target user is calculated through a weighted average of its utility to similar users (neighbors). These neighbors are users who have the highest correlation with the target user based on their liked items. Item-based CF [Sarwar et al., 2001, Xue et al., 2019], on the other hand, determines the utility of an item to a target user by assessing its similarity to other items the user has liked. The similarity between items is calculated based on the overlap in users who have liked both of them.

Model-based CF utilizes user-item interaction data to learn a predictive model, characterizing users and items by their most prominent (latent) features. Popular methods in model-based CF include matrix factorization [Koren et al., 2009, Funk, 2006], autoencoders [Sedhain et al., 2015], graph-based methods [Nikolakopoulos and Karypis, 2019], and deep learning approaches [He et al., 2017].

Collaborative filtering approaches are very effective at capturing complex usage patterns and user preferences in the absence of explicit data, and providing serendipitous recommendations through the wisdom of the crowd. However, they struggle when user-item interaction data is sparse and collaborative signals are weak. Another well-known issue with collaborative filtering is the cold-start problem, which arises with new items and users. Additionally, collaborative filtering methods can suffer from popularity bias, favoring items with numerous interactions over niche ones. They also rely heavily on the user base, which can be beneficial but may result in a lack of diversity if the user base is homogeneous.

Content-based recommenders

The idea behind content-based recommender systems is to recommend an item to a target user by matching the item’s descriptive attributes or content with the user profile (Figure 2.2) [Ricci et al., 2015, Aggarwal, 2016]. This user profile is typically created by analyzing user’s interaction history, from item features (such as tags, description, and content) or user-generated features (such as review). The item-side features can also come from the item’s content, metadata (such as tags and descriptions), or reviews written by the entire user base.

Early content-based approaches are based on keyword-matching, TF-IDF term weighting [Salton and Buckley, 1988], (latent) semantic analysis [Degemmis et al., 2007, Sarwar et al., 2000], and tag-based [Sen et al., 2009], which are closely related to established methods in information retrieval [Belkin and Croft, 1992]. More recently, deep neural networks are utilized to model users and items, such as using CNNs [Zheng et al., 2017], RNNs [Suglia et al., 2017], transformer encoders [Pugoy and Kao, 2020].

Review-based recommenders. User written reviews contain valuable information

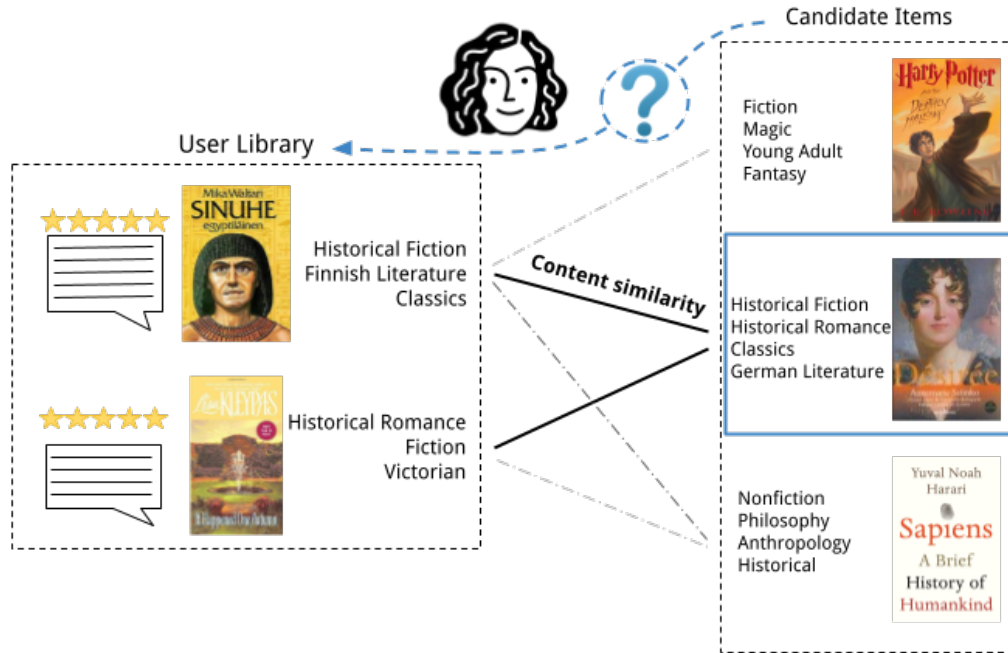


Figure 2.2: A visualization of content-based recommendation showing a user’s previous interactions in her library and their similarity to the candidate items based on matching features or content. The figure shows items similar in content to the user’s library are selected for recommendation.

that can be used to enhance all types of recommender systems [Chen et al., 2015]. Reviews are exploited as an additional data source to enhance collaborative filtering recommenders for example they are combined with rating data to mitigate data sparseness and increase interpretability [McAuley and Leskovec, 2013], or are used as regularizers [Almahairi et al., 2015]. Reviews are also helpful in context-aware [Li et al., 2010] and aspect-based [Bauman et al., 2017] recommendation by extracting users’ context and specific aspects they were interested in.

Review-based recommenders can be considered a subcategory of content-based recommender systems, increasing the novelty, diversity, and item coverage in recommendation due to utilizing the user generated information [Chen et al., 2015, Esparza et al., 2010, Zheng et al., 2017, Pugoy and Kao, 2020]. The reviews contain user-specific information indicating the reasons behind their interest, making them a better source of user profile that captures user’s personality, writing style, and subjective interests as opposed to using item’s static description which creates the same user profile for all users who liked the same item [Chen et al., 2015].

Content-based recommenders excel in text-rich and unstructured domains, gaining new momentum with the availability of new types of side information like item meta-data and user-generated content. They are more transparent and interpretable, do

not suffer from the cold-start problem for new items, and can mitigate the cold-start problem for users by requiring minimal manual input from new users. However, one disadvantage of content-based models is that they tend to provide users with obvious recommendations. This problem is more pronounced when profiles are built from limited item features, such as categories or keywords. Using user reviews can help alleviate this issue by leveraging user-specific preferences and novel aspects within user-generated content.

In this thesis, we address *sparse data regimes* with limited collaborative signals, emphasizing content-based methods while occasionally incorporating user-item interactions. Our approach focuses on personalization through **textual user profiles** derived from diverse sources, including user-item interaction histories, reviews, questionnaires, and less conventional inputs such as chat conversations.

Knowledge-based recommenders

Knowledge-based recommender systems are closely related to content-based recommenders, as both leverage item features and can utilize detailed user profiles. However, in knowledge-based recommenders, the item features are explicit and typically sourced from an external knowledge base. The recommendation process is often interactive, designed to match explicit user requirements by leveraging detailed domain knowledge and user preferences [Aggarwal, 2016]. Knowledge-based recommenders perform well in cold-start situations where user data is sparse, but as usage patterns are accumulated, learning-based methods become more effective [Ricci et al., 2022].

Hybrid recommenders

These recommender systems combine two or more of the above mentioned recommender approaches to enhance the performance by making use of each method’s advantages [Ricci et al., 2022]. A common combination is collaborative filtering with content-based recommenders, where the latter deals with cold start problem and sparse data and the former take advantage of usage patterns [Paradarami et al., 2017].

Transformer-based recommenders

Early work on using transformers for recommendation benefited from the self-attention mechanism to model sequential patterns in user-item interactions [Kang and McAuley, 2018, Sun et al., 2019]. These models were primarily ID-based and did not incorporate textual features. In contrast, a parallel research direction explored the use pre-trained (large) language models (PLMs and LLMs) in recommender systems, taking advantage of their capabilities in text encoding and their encoded world knowledge [Wu et al., 2024, Lin et al., 2023]. For example, [Pugoy and Kao, 2020, Wu et al., 2021a, Zhang et al., 2021] applied PLMs like BERT as contextual text encoders to enhance the understanding and representation of user and item textual features.

More recently, the P5 model [Geng et al., 2022] proposed a unified sequence-to-sequence framework for various recommendation tasks, including direct and sequential recommendation, rating prediction, review summarization, and explanation generation, by fine-tuning T5 model [Raffel et al., 2020] in a multi-task prompt-based setting, optimizing the same language modeling objective. Building on P5, [Hua et al., 2023] demonstrated that advanced indexing strategies, including semantic similarity and collaborative filtering, enhance the recommendation performance.

To tap into language models’ built-in world knowledge, [Penha and Hauff, 2020] probed BERT to understand its knowledge about books, movies, and music. [Hou et al., 2024] investigated the potential of LLMs as zero-shot rankers using prompts containing user history and candidate items. Their experiments highlighted LLMs may suffer from position and popularity biases, suggesting methods to mitigate these issues. Additionally, [Kang et al., 2023] examined rating prediction with LLMs across various setups, including zero-shot, few-shot, and fine-tuned configurations, finding that only fine-tuned models could match or surpass traditional recommenders.

2.1.3 Challenges

Recommender systems pose a variety of challenges in both research and industry, with different aspects gaining importance depending on the domain and application [Ricci et al., 2022, Khusro et al., 2016]. These systems are inherently *multi-stakeholder*, involving users, platforms, and other item-related stakeholders, all of whom must be considered in the design of the recommender algorithms. A lack of *fairness* can arise when the model fails to consider the interests of all stakeholders. A related issue is *popularity bias*, where popular items are favored over *long-tail* items. Another well-known challenge is the *cold-start* problem, where the introduction of new users or items into an already functioning system leads to inaccurate recommendations.

A fundamental challenge in recommender system research is in *preference acquisition*. Explicit user preferences are high quality but rare; in contrast, implicit user interaction data is more abundant but often weak and uncertain. The issue of *missing data*, whether a user has not interacted with a certain item due to dislike or simply because they were never exposed to it, is especially pronounced with implicit data. Recommender systems, especially those based on collaborative signals, suffer from *data sparsity*, which hinders the ability to identify meaningful patterns and deliver accurate recommendations.

Beyond accuracy, other key criteria such as *novelty*, *diversity*, and *serendipity* add notable value to recommendations and should be integrated in system design and evaluation. Additionally, *interpretability* and *explainability* (understanding the rationale behind the recommendations), along with *scrutability* (allowing users to scrutinize and adjust recommendations), have become increasingly important topics for users interacting with AI systems.

Ultimately, the main goal of recommender systems is to present users with truly relevant and interesting items without overwhelming them with irrelevant information, ensuring that they remain interested and engaged. Domain and application-specific requirements play a major role in finding the best strategies for a system. Examples of different domains with varied demands (considering monetary, time, and mental commitments) include music, food, movies, fashion, news, books, travel, and financial investments.

In this thesis, we address personalization in *sparse data* settings using both *explicit* and *implicit user feedback*, focusing on challenges in evaluation and training strategies of implicit feedback data. We pay special attention to *cold-start* and *long-tail* users and items. Our content-based approaches inherently avoid *popularity bias*, ensuring fair recommendations. Additionally, leveraging textual user profiles promotes *scrutability* and *transparency*, enabling users to better understand and adjust their personalized experiences.

2.2 Personalized Search

Advancements in information retrieval (IR) have enabled users to find relevant information on the web or within large data collections [Manning et al., 2008]. However, traditional IR systems return identical results for the same query, regardless of the user who issued it. Search engines consider hundreds of interleaving factors (such as text statistics, popularity, freshness, and locality) when ranking the results [Lewandowski, 2023]. Typically, non-personalized search engines retrieve documents aligned with the most popular query intent or, at best, deliver results reflecting a variety of query interpretations.

This approach proves less effective for users with diverse backgrounds and interests, especially when their queries are short and ambiguous [Wen et al., 2018]. Moreover, it fails to account for evolving user information needs over time [Chang and Deng, 2020]. For example, a query like *python* might correspond to the programming language for a computer scientist but refer to the snake for a zoologist. If the computer scientist recently watched a nature documentary, her interest could shift toward information about the python snake.

These shortcomings have led to the need for context-sensitive and personalized information retrieval (PIR), which tailors results not only to the query but also to the individual user [Ghorab et al., 2013, White, 2016]. PIR systems achieve this by leveraging additional user information and interaction history to adapt the query or reorder non-personalized search results according to the user’s unique preferences and needs.

Subsection 2.2.1 introduces the required background on general information retrieval concepts. In Subsections 2.2.2 and 2.2.3, we review the literature on personalized information retrieval and personalized entity search, respectively.

2.2.1 General Concepts of Information Retrieval

Task Definition

The most common task in information retrieval is *ad hoc retrieval*, which involves retrieving the most relevant information from a large collection of unstructured or semi-structured text, in response to a user *information need* expressed through a specific *query* [Manning et al., 2008]. The goal of IR systems is to return a *ranked* list of documents ordered by their relevance to the query.

Query Expansion

Users often express their information needs as short, non-exhaustive queries, which can negatively impact retrieval performance by limiting the system’s ability to fully capture the users’ intent. Query expansion (QE) addresses this issue by adding related terms to the original query, increasing the chances of retrieving relevant documents [Carpineto and Romano, 2012]. The added terms may include synonyms and related concepts sourced from external resources such as WordNet, terms that frequently co-occur with the original query terms in a text corpus or query logs, and terms derived through (pseudo) relevance feedback, which incorporates information from the initial retrieval results back into the query.

Relevance Feedback

Relevance Feedback (RF) enhances retrieval performance by refining the system’s understanding of user intent through iterative user feedback. After an initial query, the user labels the retrieved documents as relevant or non-relevant, and this feedback is used to adjust the retrieval process. Traditionally, relevance feedback has been closely tied to query expansion, where new terms from relevant documents are added to the query to improve recall and precision. However, RF can be used independently in re-ranking approaches, where the feedback is used to reorder the retrieved documents without altering the query itself. *Pseudo relevance feedback* (PRF) extends RF by assuming the top-ranked documents from the initial retrieval are relevant, automating the feedback process without explicit user input.

Vector Space Model

In the Vector Space Model (VSM), both documents and queries are represented as sparse weight vectors in the same high-dimensional space, where each dimension corresponds to a vocabulary term [Salton et al., 1975]. The relevance score between a document and a query is then calculated by *dot product* or *cosine similarity* between their corresponding vectors.

TF-IDF Weighting. One of the most effective and used weighting scheme for the VSM is *TF-IDF* [Salton and Buckley, 1988]. TF (Term Frequency) refers to

the number of times the term occurred in the document. IDF (Inverse Document Frequency) is the inverse of the number of documents that the term appeared in in the entire collection, giving less importance to very common terms that appear in many documents.

Probabilistic models

IR systems have uncertainty about the relevance of each document to the given query due to limited information about user intent, document content and the relationship between them. The known properties of documents and the query provide, at best, a probabilistic evidence of relevance. Probabilistic approaches to IR estimate the likelihood of a document being relevant to a given query based on probability theory and rank the documents in descending order of estimated probability of relevance to the query [Maron and Kuhns, 1960, Jones et al., 2000, Robertson and Zaragoza, 2009]. Binary relevance assumption holds in the *Probability Ranking Principle*, meaning a document is either relevant to an information need or it is not.

$$\text{score}(q, d) = P(R = 1|d, q) \propto \frac{P(R = 1|d, q)}{P(R = 0|d, q)} \propto \frac{P(d|R = 1, q)}{P(d|R = 0, q)}$$

where R is an indicator random variable determining the relevance of document d to the query q .

Binary Independence Model (BIM). This is one of the foundational probabilistic approaches in IR with two assumptions that simplify estimation of $P(R|d, q)$. First, it assumes a binary representation of term occurrence, indicating the existence or absence of a term in the document and query. Second, it assumes that the occurrence of each term in the document is independent of the occurrence of other terms, enabling a more straightforward probabilistic calculation of relevance. [Robertson and Jones, 1976] estimate the relevance under the aforementioned assumptions as:

$$\text{score}(q, d) = \sum_{w \in V_q \cap V_d} \log \frac{(r_w + 0.5)(N - R - n_w + r_w + 0.5)}{(n_w - r_w + 0.5)(R - r_w + 0.5)}$$

where $V_q \cap V_d$ is the intersection of query and document vocabulary, N is number of judged documents, n_w is number of documents containing term w from judged set, R is the number of relevant documents, r_w is number of documents containing term w from relevant set.

BM25. Okapi BM25 [Robertson and Zaragoza, 2009] is a widely used ranking function in IR that combines probabilistic methods with heuristics, demonstrating strong empirical results. Instead of the binary representation of term occurrence, it considers the frequency of query terms in documents, with reverse term frequency to control the frequent term effects and length normalization preventing longer documents from unfairly getting high scores. For a query q and the document d , the BM25 score is defined as:

$$score(q, d) = \sum_{w \in V_q} idf(w) \cdot \frac{tf(w, d) \cdot (k_1 + 1)}{tf(w, d) + k_1 \cdot \left(1 - b + b \cdot \frac{|d|}{avgdl}\right)}$$

where V_q is the query vocabulary, $tf(w, d)$ is the number of times word w appeared in document d , $|d|$ is the length of the document at hand, $avgdl$ is the average length of all documents in the corpus, and k_1 and b are tuning parameters.

The $idf(w)$ indicating the inverse document frequency of the term w , derived from BIM, is defined as:

$$idf(w) = \log \frac{N - df(w) + 0.5}{df(w) + 0.5}$$

where N is the total number of documents in the corpus, $df(w)$ is the number of documents containing term w .

Statistical Language Models

Another line of probabilistic models for information retrieval are statistical language modeling approaches with good empirical performances and solid statistical foundations [Zhai, 2008, Manning et al., 2008]. The basic idea is that the user query should have high likelihood of being generated from the language model of the relevant document, also known as the *query likelihood model*. Early work, such as in [Ponte and Croft, 1998], used a multiple Bernoulli model to compute query likelihood. However, the multinomial model [Miller et al., 1999] is more commonly used because it incorporates term frequency, in both query and document, more naturally than in the former model, resulting in a better representation of term importance [Zhai, 2008]. The query likelihood model is formulated as:

$$score(q, d) = P(q|\theta_d) = \prod_{w \in q} P(w|\theta_d)$$

where θ_d is the document language model.

Probability of a word in a document $P(w|\theta_d)$, i.e., document language model, is estimated using maximum likelihood estimation (MLE):

$$P(w|\theta_d) = \frac{tf_{w,d}}{|d|}$$

where $tf_{w,d}$ is frequency of term w in document d , and $|d|$ is document length.

Smoothing. When estimating the document language model, data sparseness is a major problem that can lead to inaccurate probability estimations. In particular, probabilities for terms that occur only once in the document may be overestimated, while query terms that do not appear in the document at all receive zero probability, which negatively affects retrieval performance [Manning et al., 2008]. To address

this, *smoothing* techniques are applied to adjust the estimated probabilities, mitigating these issues and improving retrieval effectiveness [Zhai and Lafferty, 2001]. One commonly used technique is Dirichlet prior smoothing, interpolating the estimated probability with a background language model:

$$P(w|\theta_d) = \frac{tf_{w,d} + \mu p(w|\theta_C)}{|d| + \mu}$$

where θ_C is the background language model estimated from a large corpus, and μ is the smoothing parameter.

N-grams. The n-gram language model is a probabilistic model that estimates the probability of a word given the $n - 1$ preceding words, assigning probabilities to the entire sequence [Jurafsky and Martin, 2009]. Unigrams ($n = 1$) represent the simplest form of this model, where each word's probability is considered independently of any preceding words. Higher-order n-grams such as bigrams ($n = 2$) and trigrams ($n = 3$) allow the model to incorporate a limited context by conditioning the word's probabilities on the one or two immediate preceding words, respectively. A natural way to estimating the n-gram document language model is maximum likelihood estimation (MLE), by dividing the frequency of a sequence of length n by the frequency of its prefix of length $n - 1$. For example for biragms it is calculated as follows:

$$P(w_i|w_{i-1}) = \frac{\text{count}(w_{i-1}w_i, d)}{\text{count}(w_{i-1}, d)}$$

where $\text{count}(w_{i-1}w_i, d)$ is the count of sequence $w_{i-1}w_i$ in the document d and $\text{count}(w_{i-1}, d)$ is the frequency of the prefix w_{i-1} in the document d . Similar smoothing techniques, as discussed above, can be applied to the probability estimation for n-grams.

Kullback–Leibler Divergence Retrieval Model. One limitation of the query likelihood model is its assumption that queries are generated from the document model, which makes it theoretically difficult to support expanded or modified queries. An alternative approach is to estimate language models for both the query and the document, then comparing these models using the Kullback-Leibler (KL) divergence [Lafferty and Zhai, 2017]. KL divergence is an asymmetric measure that quantifies the information loss when approximating one probability distribution with another.

$$\text{score}(q, d) = -KL(\theta_q||\theta_d) = - \sum_{w \in V} p(w|\theta_q) \log \frac{p(w|\theta_q)}{p(w|\theta_d)}$$

where θ_q and θ_d are query and document language models, and V is the vocabulary. The KL divergence score indicates how well the document model approximates the query model [Zhai, 2008]. Lower divergence values suggesting better alignment between the two models, therefore the score between the document and the query is set to the negative divergence.

Neural Retrieval Models

Learning to Rank (LTR) approaches [Li, 2011], which leverage machine learning, have gained prominence by utilizing hand-crafted features such as term frequencies, document lengths, and document source quality. A major challenge of LTR is its reliance on feature engineering. Deep learning approaches address this limitation by learning directly from raw text, reducing the dependency on manually designed features [Guo et al., 2020].

Neural IR methods can be divided into pre-transformer and post-transformer models. Pre-transformer models [Mittra and Craswell, 2018], which include feedforward neural networks (FNNs), convolutional neural networks (CNNs), and recurrent neural networks (RNNs), faced limitations in capturing long-range dependencies and complex query-document interactions. Post-transformer models [Lin et al., 2021], such as BERT-based architectures, have revolutionized neural retrieval by leveraging self-attention mechanisms to model complex relationships between terms and contextualize term semantics. Additionally, post-transformer models benefit from pretraining on large-scale textual corpora, enabling them to learn inherent patterns of human language, which significantly enhances retrieval effectiveness.

Neural ranking models can be broadly classified into two categories: *representation-based* and *interaction-based*, also referred to as *bi-encoder* and *cross-encoder* models. Representation-based models learn dense representations of both the query and the document, comparing them using a simple matching function, such as cosine similarity or dot product. Various neural architectures are employed to learn these representations, including FNNs [Huang et al., 2013], CNNs [Hu et al., 2014, Shen et al., 2014], RNNs [Palangi et al., 2016, Wan et al., 2016], and transformers [Karpukhin et al., 2020, Xiong et al., 2021]. Due to their efficiency, representation-based models are particularly well-suited for early-stage retrieval, as document representations can be precomputed and stored in advance.

On the other hand, interaction-based methods model term-level interactions between the query and document, usually capturing better matching signals. Examples of such methods include DRMM [Guo et al., 2016], KNRM [Xiong et al., 2017], PACRR [Hui et al., 2018], and transformer-based models [Nogueira and Cho, 2019, Yang et al., 2019, Nogueira et al., 2020]. Interaction-based models are more computationally intensive because they require real-time encoding of query-document pairs. They are better suited for re-ranking a smaller set of candidate documents after initial retrieval. Late interaction models, such as ColBERT [Khattab and Zaharia, 2020], bridges the gap between the two paradigms by first encoding queries and documents independently, then performing term-level comparisons at retrieval time.

The representation-based neural retrieval methods discussed above are examples of *dense retrieval*, where models learn low-dimensional dense vector representations of queries and documents [Lin et al., 2021]. In contrast, traditional retrieval models like BM25 and Vector Space Model (VSM) are examples of *sparse retrieval* methods,

representing queries and documents as high-dimensional sparse vectors, with each dimension corresponding to a unique vocabulary term. The main limitation of sparse methods lies in their lack of semantic understanding due to reliance on exact keyword matching.

Sparse retrieval benefits from efficient *indexing* techniques using inverted indexes, which map terms to lists of relevant documents, enabling fast and scalable retrieval even for large collections. On the other hand, dense retrieval processes are computationally intensive, typically managed using approximate nearest-neighbor search libraries like FAISS [Johnson et al., 2021].

Recently, *learned sparse retrieval* approaches have emerged, combining the strengths of both paradigms [Bai et al., 2020, Formal et al., 2021]. These methods learn sparse representations where non-zero vector elements correspond to vocabulary terms, leveraging the semantic understanding of neural models while maintaining the interpretability and indexing efficiency of traditional sparse methods.

Generative information retrieval [Najork, 2023] is an emerging paradigm in IR [Metzler et al., 2021]. Unlike traditional methods that rely on stored document indexes or vector embeddings to retrieve documents, these systems encode documents directly into the model’s internal memory, enabling them to generate document identifiers or even content directly [Zhu et al., 2023]. However, generative IR models face critical challenges, such as mitigating hallucination and ensuring faithfulness to the original document sources.

2.2.2 Personalized Information Retrieval

Personalized information retrieval (PIR) systems leverage user data to tailor search results, targeting either an individual user or a community [Ghorab et al., 2013]. Selective personalization plays a crucial role in web search, as not all queries or search contexts benefit from personalization [Teevan et al., 2008, Bennett et al., 2015]. When used appropriately, personalization enhances the search experience by reducing the effort users need to invest in refining their queries or scrolling through results to find subjectively relevant information.

PIR systems consist of two primary components: one to construct the *user model or profile* and another utilizing it to personalize search results [Ghorab et al., 2013, Liu et al., 2020b]. The user modeling process involves collecting and representing user data, with a focus on the types and sources of information used for personalization. User information can be explicitly provided by users [Gauch et al., 2007], for instance, by inquiring their topics of interest and enriching their profiles using their explicit relevance feedback on documents [Micarelli and Sciarrone, 2004].

Implicitly gathered data, however, has been more extensively studied due to its zero user-side effort and potential for large-scale collection. One of the most prominent features studied is user search behavior, particularly *query-click logs* [Speretta and

Gauch, 2005, Silvestri, 2010]. Beyond click-through data, browsing behavior such as dwell time, scrolling patterns, and mouse movements has also proven valuable [Agichtein et al., 2006]. For example, the content of web pages visited by users over extended periods was used to build user profiles in [Matthijs and Radlinski, 2011]. Additionally, [Teevan et al., 2005] demonstrated that rich user profiles, comprising both search-related data and other user information like email messages, calendar events, and stored documents, significantly enhanced personalization. User geo-location and language are other key signals for personalizing search, having been integrated into modern search engines [Bennett et al., 2011]. Other sources of user data for search personalization include folksonomy data (i.e., user-generated tags in social bookmarking communities) [Zhou et al., 2017, Biancalana et al., 2013], user desktop files [Chirita et al., 2007], and user-specific demographic attributes such as age, gender, marital status, interests, and occupation [Cheng and Cantú-Paz, 2010]. Recent studies have also employed machine learning approaches to improve user modeling. For instance, [Vu et al., 2017] learned user embeddings within a topical interest space, leveraging Latent Dirichlet Allocation (LDA) for query and document representation.

In addition to long-term user behavior, short-term in-session user context has been explored for *session-based personalization* [Shen et al., 2005]. Further research by [Bennett et al., 2012] examined the interplay between long-term and short-term user behaviors, highlighting how they can be leveraged to personalize results at different stages of the search process. [Ge et al., 2018] modeled both in-session and long-term user behaviors using hierarchical RNNs to create dynamic, query-aware user profiles. Similarly [Zhou et al., 2020] employed a hierarchical transformer to model queries along with short-term and long-term user histories; they also explored building a personalized language model to further disambiguate queries based on user context.

The second component of PIR involves applying personalization using the user model. Two primary approaches for this task are: *query expansion* or *reformulation*, and *re-ranking* the search results [Ghorab et al., 2013, Liu et al., 2020b]. Query expansion works by adding user-specific information from the user profile to the query for personalized retrieval [Chirita et al., 2007, Zhou et al., 2017, Biancalana et al., 2013]. Alternatively, [Yao et al., 2020] explored query reformulation using personal word embeddings to disambiguate query terms. In addition to web search, query expansion has been applied in other domains, such as personalized email search [Kuzi et al., 2017].

Result re-ranking approaches reorder the top-k non-personalized retrieved results based on the user model. For example, [Teevan et al., 2005] created user profiles as a form of relevance feedback and modified the BM25 ranking function to incorporate it. [Sontag et al., 2012] applied personalization by incorporating user-specific priors into a probabilistic language modeling approach. [Wang et al., 2013, Song et al., 2014] adapted generic neural rankers to each individual user using series of linear transformations to adjust the global ranking model parameters based on limited user data. More recently [Zhou et al., 2021] explored the benefits of self-supervised

pre-training in a two-stage training framework to learn better representations for the downstream personalized ranking task.

Another successful area of research in information retrieval is *query auto-completion* [Cai and de Rijke, 2016b], where the system suggests alternative ways to extend the entered prefix to form a complete query. Personalized query auto-completion improves effectiveness by leveraging both the user’s short-term and long-term context. For example, [Shokouhi, 2013] utilized demographic features (age, gender, and location) along with user-specific features (short-term and long-term search history) to train an auto-complete reranker. They demonstrated that certain features, such as geo-location and long-term search history, proved more effective than others. [Cai and de Rijke, 2016a] proposed a selective personalization approach that determines when to apply personalization based on factors like the typed prefix, clicked documents, and preceding queries. In this thesis, however, the focus is on item ranking rather than suggested query ranking.

2.2.3 Personalized Entity Search

An *entity* is defined as “a uniquely identifiable object or thing, characterized by its name(s), type(s), attributes, and relationships to other entities” [Balog, 2018]. Research in this area encompasses various tasks related to entities [Balog et al., 2010], but in this subsection, we focus specifically on the scenario where the user is searching for an entity or a list of entities, and the retrieval system returns a ranked list of entities rather than documents. This task is known as *entity retrieval*. Non-personalized entity search has been a long-standing area of research [Huang et al., 2007, Dietz, 2019]. Recently, approaches such as [Gerritse et al., 2022] have enhanced BERT by integrating entity embeddings from a knowledge graph, mapping these entities into the same space, and injecting them into the BERT model to improve performance.

Research on personalized entity search has mostly been limited to the entertainment and e-commerce (product) domains, particularly the latter due to the rapid growth of e-commerce. CLEF social book search competition [Koolen et al., 2016] explored the complexity of relevant book retrieval for complex user information needs, utilizing various types of information such as tags, reviews, posts, and ratings from large book communities like LibraryThing and Amazon. However, the distribution of the developed data collection has since been discontinued.

In the e-commerce domain, [Jannach and Ludewig, 2017] explored personalized search using recommendation techniques. [Ai et al., 2017] focused on learning user and item embeddings within a joint latent space, leveraging user-written reviews. Similarly, [Zhang et al., 2020] proposed an efficient and scalable approach to product search through embedding learning. Recently, the joint training of search and recommendation tasks has gained interest. For instance, [Zamani and Croft, 2020, Penha et al., 2024] developed a unified model for both search and recommendation in a multi-task learning framework, enhancing the performance of both tasks.

This thesis aligns most closely with research in personalized entity search, where the user initiates a query and the system returns a ranked list of entities or items that are relevant to the query and tailored to the user’s preferences and interests. We refer to this task as **search-based recommendation**, exploring personalization across various domains by utilizing both the user profile and contextual search queries to assess relevance.

2.3 User Profiling

The increasing adoption and success of personalized services such as recommender systems (Section 2.1) and personalized search (Section 2.2) have driven significant research interest in *user profiling and modeling*. This field explores various aspects of user information, including the types and sources of data, collection strategies, and model representation techniques [Eke et al., 2019, Gauch et al., 2007, Purificato et al., 2024].

A notable area of research is *personal information management* (PIM), which focuses on facilitating the control, search, and organization of personal data [Jones, 2007, Montoya et al., 2018]. Along similar lines, [Balog and Kenter, 2019] proposed a research agenda for building a *personal knowledge graph* (PKG), comprising entities, attributes, and relations that are personally relevant to the user. Elements of PKGs may be linked to public knowledge graphs or repositories like Wikidata and IMDb. For instance, [Yen et al., 2019] constructed PKGs from users’ tweets, while [Chakraborty et al., 2022] created PKGs incorporating research-related information such as collaborators, affiliations, and research interests.

Generic user models focus on creating general-purpose, domain-independent and task-agnostic representations of users [Kobsa, 2007]. Recent approaches aim to develop universal user models suitable for multi-task settings [Ni et al., 2018] or lifelong learning frameworks capable of continual updates without forgetting prior knowledge [Yuan et al., 2021]. Holistic models, such as the one proposed by [Musto et al., 2020a], aggregate user data from heterogeneous sources like social networks, wearables, and smartphones, enabling personalization tasks [Musto et al., 2020b].

Most personalized systems rely on *domain-specific user profiles*, tailored to specific tasks or applications. These profiles are designed to capture users’ preferences and behaviors within specific contexts, such as e-commerce, news, or healthcare. *User behavior modeling* plays a central role in these systems by analyzing users’ direct interactions with platforms and services. For example, click-through data has been widely used to model user behavior [Zhou et al., 2018, Silvestri, 2010]. Building on behavioral data, [Wang et al., 2018] utilized users’ click histories to propagate interests through a knowledge graph enriched with auxiliary information. Multi-behavior modeling, which includes actions like clicks, likes, add-to-cart events, and purchases, has been explored by [Jin et al., 2020]. *Ontological user profiles*, created using domain-

specific concepts and semantics, have also been employed in personalized search [Sieg et al., 2007] and recommendations [Middleton et al., 2004].

One of the earliest approaches to personalization, which remains relevant today, is *demographic-based profiling* [Al-Shamri, 2016]. These methods leverage demographic attributes such as age, gender, occupation, and location to enable personalization, relying on correlations between users with similar features. Advances in inferring users’ demographic attributes from their online traces [Preotiuc-Pietro et al., 2015, Tigunova et al., 2019] shows the viability of creating such user profiles automatically and in a large scale.

An emerging trend in personalization research is using *human-comprehensible user profiles* to improve system *transparency* and *scrutability*, enabling users to understand and critique the personalization process [Radlinski et al., 2022]. *Textual user profiles*, represented in natural language or as sets of tags, express these characteristics. In [Balog et al., 2019], user preferences are modeled as a weighted set of tags presented to users in natural language through fixed templates, allowing them to select statements that best describe their interests. Similarly, [Mysore et al., 2023] create user profiles as a set of weighted concepts, offering users more flexibility by enabling textual edits to these concepts.

Natural language profiles have also been explored in LLM-based recommender systems, leveraging their deep language understanding and extensive world knowledge for tasks such as conversational recommendation, rating prediction, sequential recommendation, and item ranking. [Sanner et al., 2023] demonstrated that user-provided text profiles, gathered in a user study, enable LLM-based recommenders to perform competitively when ranking a small set of items, particularly for cold user data. Similarly, [Ramos et al., 2024] investigated this effect on warm user data by automatically generating user profiles from a selection of user reviews, using another LLM, and evaluated their approach on the rating prediction task. Although the performance in both studies falls short of state-of-the-art recommenders, they highlight a promising direction for further research in this area.

In this thesis, we focus on **textual user profiles** for personalization, emphasizing transparency and scrutability for users. Our work examines both ends of the spectrum: user-created profiles that require significant effort to maintain and profiles generated implicitly from users’ online activities. We investigate various sources for creating these profiles, examining techniques to expand them for maximum utility, while also exploring methods to constrain them, reducing noise and computational costs.

2.4 Evaluation

The primary evaluation criteria in personalized information retrieval (Section 2.2) and recommender systems (Section 2.1) research are accuracy-based measures, which

assess the utility of returned results in terms of their *relevance* and *interestingness* to the user [Ricci et al., 2015, Manning et al., 2008]. Beyond accuracy, additional metrics such as novelty, diversity, and serendipity have also been explored for evaluating these systems [Kaminskas and Bridge, 2017]. However, these criteria are beyond the scope of this thesis.

User assessments can take two forms: graded or binary judgments. Graded judgments are typically gathered explicitly, while binary assessments can be collected either explicitly or implicitly, such as through user clicks. In a rating prediction model, where the system predicts the users’ item rating, the most common metrics calculate the error between the predicted and ground truth assessments via *Mean Squared Error (MSE)* and *Mean Absolute Error (MAE)*. However, the focus of this thesis is on ranking models, where the system outputs a list of ranked items. Therefore, ranking metrics are the most suitable for measuring the performance of such a system.

2.4.1 Metrics

NDCG@k. Normalized Discounted Cumulative Gain (NDCG) is designed to measure the effectiveness of a ranking system by evaluating the relevance of retrieved items at each rank position. It discounts the score if a relevant item appears in a lower rank rather than at the top, aligning with user behavior that prioritizes top-ranked results. While NDCG can handle graded relevance, it is also applicable in binary relevance settings. The Discounted Cumulative Gain (DCG) is calculated as:

$$DCG@k = \sum_{i=1}^k \frac{2^{rel(i)} - 1}{\log_2(i + 1)}$$

with $rel(i)$ indicating the relevance of the item at rank i in the predicted ranked list.

The NDCG is calculated as the ratio of the DCG to the Ideal DCG (IDCG):

$$NDCG@k = \frac{DCG@k}{IDCG@k} \quad IDCG@k = \sum_{i=1}^{|rel_k|} \frac{2^{rel(i)} - 1}{\log_2(i + 1)}$$

where rel_k is the list of relevant items up to the position k . The normalization allows the NDCG of different samples to be comparable by dividing by the highest possible score (IDCG) per sample. The final NDCG value is the average of NDCGs for each sample.

MRR. Mean Reciprocal Rank (MRR) is another metric designed to evaluate ranking systems, especially in cases where there is only one relevant document. The reciprocal rank computes the inverse rank of the first relevant item in the list, and MRR is the average of reciprocal ranks for the entire dataset:

$$MRR@k = \frac{1}{N} \sum_{i=1}^N \frac{1}{rank_i}$$

where N is the number of instances in the dataset, and $rank_i$ is the rank of the first item in the list that was relevant to the user.

P@k. Precision is the ratio of relevant documents to the retrieved ones, and is a set-based evaluation metric that does not consider ranking. Precision@k ($P@k$) is an extension designed for ranked lists, measuring the proportion of relevant documents among the top- k returned results.

$$P@k = \frac{\#tp_k}{k}$$

where $\#tp_k$ is the number of relevant items in the top- k .

When the number of ground-truth relevant documents is smaller than k , even the perfect ranking does not achieve $P@k$ of 1, making a small cut-off such as $k = 1$ a common metric to be used.

R@k. Recall is the percentage of the relevant documents that are retrieved, without any consideration for the ranking. Recall@k ($R@k$) is the ratio of relevant documents among the top- k to the total number of relevant documents.

$$R@k = \frac{\#tp_k}{R}$$

where $\#tp_k$ is the number of relevant items in the top- k , and R is the number of all relevant documents.

The denominator of the $R@k$ can be larger than k , therefore even the perfect ranking may not achieve recall@k of 1. Some studies such as [Liang et al., 2018] used an alternative formula for $R@k$ that replaces the denominator with $\min(k, R)$, however, this is not widely used.

Other metrics. In addition to the discussed metrics, MAP and AUC are among the commonly used metrics in evaluating recommender systems. MAP (Mean Average Precision) calculates the average of $P@k$ at different cut-offs where relevant documents appear in the ranking. In ROC AUC (Receiver Operating Characteristic Area Under Curve), the True Positive Rate (TPR) is plotted against the False Positive Rate (FPR) at various thresholds, and the area under this curve is reported.

2.4.2 Significance Testing

In order to draw reliable conclusions about the results of an experiment and reduce the risk of interpreting random variation as meaningful differences, we perform sta-

tistical significance testing. These tests help determine whether observed differences in evaluation metrics, such as $P@k$, are statistically significant or due to chance.

Paired t-test. The paired t-test is a statistical method used to compare the means of two related sets of samples, such as evaluation scores for two systems tested on the same set of queries. The null hypothesis assumes that the mean difference between paired observations is zero, where paired observations represent performance metrics of two models evaluated on the same dataset.

Given the computed metric for each sample (here query or user-query pair) from the two models, we calculate the mean (\bar{d}) and standard deviation (s_d) of the differences between paired scores. The t-statistic is then computed as:

$$t = \frac{\bar{d}}{s_d/\sqrt{n}}$$

where n represents the number of samples (e.g., queries or user-query pairs). The p-value is obtained by referencing the t-distribution with $n - 1$ degrees of freedom. If the p-value is smaller than a predefined significance level (e.g., $\alpha = 0.05$), the null hypothesis is rejected, indicating that the two models perform significantly differently.

Bonferroni correction. When multiple hypotheses are tested, the probability of falsely rejecting a null hypothesis (Type I error) increases. The Bonferroni correction adjusts the significance level (α) to α/m , where m is the number of tests.

Other significance tests. The Wilcoxon signed-rank test is a non-parametric alternative to the paired t-test, comparing the median differences of paired observations and providing robustness to outliers and non-normal distributions. For multiple hypotheses testing, ANOVA or the Friedman test, which are generalizations of the Student's t-test and Wilcoxon signed-rank test respectively, can also be used.

Chapter 3

Sparse and Scrutable User Profiles

Contents

3.1	Introduction	34
3.2	Methodology	36
3.2.1	Search-based Recommendation	37
3.2.2	User Profiling	37
3.2.3	Re-ranking Methods	38
3.3	Data Collection	40
3.3.1	Queries and Documents	40
3.3.2	User Study	41
3.4	Experiments	41
3.4.1	Research Questions	41
3.4.2	Experimental Setup	42
3.5	Results	43
3.5.1	Main Findings	43
3.5.2	Ablation Study	44
3.6	Related Work	45
3.7	Conclusion	46

In this chapter, we investigate concise user profiling in the search-based recommendation paradigm. More specifically, we aim to capture user preferences from a short questionnaire filled out by the user, such that the resulting user profile can be shown to the user as explicit text and users can easily comprehend and scrutinize it. This effort is in contrast to the data-hungry approaches that create latent representations from users' click logs and interaction histories. Our experiments show that even very sparse information about individuals can enhance the effectiveness of search results.

3.1 Introduction

Motivation and Problem. The information overload on the web has led to increased reliance on personalization in online services and platforms to keep users satisfied and engaged. However, lack of transparency is reducing user trust in system. A growing demand among users is to understand what type of data is captured from their online traces and to have the ability to review and edit their online *profiles*. This has driven emerging research in the areas of *scrutability* and *interpretability* of recommender systems. The fewer traits the profile contains and the more explicit they are (as opposed to learned latent models), the more scrutable and actionable the personalization model becomes from a user perspective.

The use of *click logs* and *user-item interactions* has led to successful models for *personalized search* and *recommender systems*. In contrast to this data-hungry paradigm, the focus of this work is on minimal data usage to strengthen the users’ ability to understand and control their data and profiles.

Personalization is important across various applications including AI assistants and chatbot, search and information access, and recommender systems. Among these, a realistic and challenging use case is *search-based recommendation*, as it requires not only interpreting the user’s intent from a text query but also integrating this with the user’s profile and history.

We investigate the role of questionnaire-based user profiles for search-based recommendation, with the challenging case that the only per-user knowledge is a sparse profile obtained from a short questionnaire.

State of the Art and its Limitations. Personalization to improve web search result ranking has been a long-standing theme in information retrieval [Wen et al., 2018]. The most important line of exploiting user information for general web search is based on *query-and-click logs* (e.g., [Teevan et al., 2005, Silvestri, 2010]). This helps in interpreting user interests and intents for ambiguous queries, as well as for identifying salient pages for popular queries. In addition to the query logs, information about the user’s location, language, and daytime are major assets (e.g., [Bennett et al., 2011]).

Recommender systems is another well studied area of personalization for ads, news, entertainment services, and e-commerce products (e.g., [Ricci et al., 2015, Gomez-Uribe and Hunt, 2016, Smith and Linden, 2017]). Here, structured data about users’ online behavior is leveraged, most notably, clicks on recommended items, ratings and purchases of products, likes of photos and videos, etc. However, these approaches are at least as data-intensive as the search engines, and require extensive user-specific data. This field has recently paid attention to *scrutable recommendations* that are comprehensible by end-users and pinpoint the specific data that explains how the recommended item was computed [Zhang and Chen, 2018, Balog et al., 2019, Pei et al., 2019].

Entity search about people, products or events has received great attention and has been incorporated into major search engines (see, e.g., [Bast et al., 2016, Balog, 2018] and further references there). This methodology leverages large knowledge graphs to infer the focus of the query and/or return crisp entities as answers. However, except for special cases such as music recommendation [Carterette, 2019] and consumer product search [Ai et al., 2017], there is hardly any work on *personalized* entity search with individual user traits.

User profiling lies at the core of all personalization tasks. [Ni et al., 2018] explored the concept of learning a universal user model applicable to multiple personalization tasks, while [Balog and Kenter, 2019] proposed building a personal knowledge graph, which can be used in recommendation [Yang et al., 2022].

Approach and Contributions. This chapter explores the direction of search-based recommendation, relying solely on a user-provided concise profile for scrutability. The requirement is that users can fully understand and control the information that drives the personalization. This includes the ability to modify or revoke pieces of a profile. Moreover, such a profile should be as sparse as possible while still giving benefits. We consider online *questionnaires* as a source of sparse and scrutable user profiles and use a set of re-ranking methods to personalize the results.

To address the search-based recommendation problem, we adopt a two-stage retrieval and re-ranking paradigm. In the first stage, items (documents) relevant to the query are retrieved using a standard search engine (site-restricted to a product platform), resulting in non-personalized results. In the second stage, the pool of candidate items is re-ranked with respect to the user profile, producing personalized results. We cast the users’ background knowledge into user-specific language models or queries. Rather than taking entities mentioned in the user’s answers (e.g., favorite books, movies, or singers/bands) in surface form, we incorporate entity descriptions from knowledge graphs as additional inputs. We also utilize word embeddings (e.g., from word2vec) for improved semantic matching. In this way, we derive a suite of *re-ranking methods*.

The key hypothesis that we test in this study is that even a very small profile about a user can improve the quality of search results as assessed by the users themselves. To this end, we hired MTurk workers for a two-stage user study. First, we collected user profiles through a questionnaire. Then, the same MTurk users assessed the interestingness of items. In total, we evaluated 115 user-query pairs (more than 2000 assessed items) from 33 distinct users.

Our experiments compare a variety of re-ranking methods, with different degrees of incorporating sparse user profiles. Our findings indicate that even sparse profiles yield statistically significant benefits over not personalizing at all.

Example: To better illustrate our work, Figure 3.1 shows an example questionnaire obtained from the crowd workers at Amazon MTurk. It captures demographic at-








Demographics		
What is your age?	<input checked="" type="checkbox"/>	31-50 years old
What is your gender?	<input checked="" type="checkbox"/>	Female
Where are you from?		New York, USA
What is your family status?	<input checked="" type="checkbox"/>	Married
Preferences		
What are your hobbies?		Reading, Baking and walking
Please name 3 of your favorite books.		Catcher in the Rye Love Poems by Pablo Neruda The Celestine Prophecy
Which kind of books do you like?		Poetry, science fiction, true crime
Please name 3 of your favorite movies/series.		Return of the King The Hobbit Requiem for a Dream
Which genre of movies do you like?		Science Fiction, documentary
Please name your favorite singers/bands/composers.		Blink 182, Portishead

Figure 3.1: Sparse user profile from questionnaire.

tributes (e.g., age, gender, location), personal tastes regarding books, movies, and music, as well as hobbies—all entered as free-form text (as opposed to guiding users through menus, which may introduce bias). The questionnaire consists of only 10 questions, and users spent 7 minutes, on average, to fill in their answers.

Consider a user, shown in Figure 3.1, who is looking for a new book about “time travel”. Figure 3.2 shows three results for the query “time travel”, along with the user’s judgments and justification sentences. As shown in Figure 3.1, she is from “New York”, and one of her favorite books is “Love poems from Pablo Neruda”. Our personalized re-ranking correctly infers that she would find an emotional book “very interesting” and a book with a New-York-based storyline “interesting”.

3.2 Methodology

This section discusses how we address search-based recommendation using sparse questionnaire-based user profiles. The first subsection describes the formulation of the search-based recommendation problem. In the second subsection, we explain the formats and characteristics of the questionnaires and detail how user input from the questionnaires is cast into user models, optionally enriched by the entities they



Figure 3.2: Examples of search results for the query “time travel” and the user’s judgements and justifications.

contain. In the third subsection, we present how user models are utilized for personalized item recommendation using various rankers, including statistical language models, BM25, and neural methods.

3.2.1 Search-based Recommendation

The search-based recommendation problem consists of a query, a user, and a list of items that should be both relevant to the query and interesting to the user. Queries are medium-grained—not too broad, such as genre categories, and not overly specific, such as searching for a particular item. The results are entity-level documents that refer to a specific item, such as a book, movie, or product. We refer to the results as “items” or “documents” interchangeably, avoiding the term “entities” to prevent confusion with more general named entities like people and locations.

This problem can be formulated as a two-stage retrieval and re-ranking model. In the first stage, given the query, the system retrieves items (entity-level documents) that are relevant to the query. The results from this stage form a non-personalized candidate item pool, which is then passed to the second, re-ranking stage. In the second stage, the candidate items are re-ranked based on the user profile, optionally taking the query into account as well.

3.2.2 User Profiling

In contrast to prior works that personalize search results using extensive logs of user queries, clicks, and other activities, we focus on sparse and concise models of user-specific interests and tastes. To this end, we limit the source of user profiling to a short questionnaire filled by the users. As shown in Figure 3.1, our designed questionnaire

covers basic demographics and personal attributes, such as hobbies, favorite books and book genres, favorite movies and movie genres, and favorite singers or music bands. The advantage of this minimal profiling approach is that the resulting profile is easily comprehensible to the user and easy to control. Users can update their profile as their interests evolve or if they have privacy concerns. This level of scrutability and controllability is not achievable with large-scale logs or latent models derived from massive user data.

As most fields in the questionnaire consist of free-form text, we treat the profiles as text documents. Depending on the ranking method employed, these profiles can be modeled as a statistical language model, a bag-of-words model, or a term-sequence model.

Enriching User Models by Entities. User profiles may include named entities, such as favorite books, movies, and musicians, as well as conceptual entities, such as book and movie genres, and hobbies. To take advantage of the sparse data to its fullest potential, we employ entity disambiguation to link entity mentions to their respective entries in a knowledge base. We then enrich the user models (language model or bag-of-words model) with descriptions of the linked entities obtained from the knowledge base, which could be based on Wikipedia or specialized knowledge repositories.

An example of entity description is shown in Figure 3.3 for one of the user’s favorite movies “Requiem for a Dream”. Typically, the description provides a summary of the movie’s story, along with additional details such as the cast and director.

Requiem for a Dream is a 2000 American [psychological drama](#) film directed by [Darren Aronofsky](#) and starring [Ellen Burstyn](#), [Jared Leto](#), [Jennifer Connelly](#), [Christopher McDonald](#), and [Marlon Wayans](#). It is based on the 1978 [novel of the same name](#) by [Hubert Selby Jr.](#), with whom Aronofsky wrote the screenplay. The film depicts four characters affected by [drug addiction](#) and how it alters their physical and emotional states. Their addictions cause them to become imprisoned in a world of delusion and desperation. As the film progresses, each character deteriorates, and their delusions are shattered by the harsh reality of their situations, resulting in catastrophe.

Figure 3.3: Description for movie “Requiem for a Dream”, from its Wikipedia page.

3.2.3 Re-ranking Methods

Given a pool M of non-personalized results for a query q and a user u , we want to re-rank the results such that the ranking reflects the user’s individual interests. All re-ranking methods treat candidate items as text documents.

Statistical Language Models

This method adopts a query-likelihood model, where the score of document $d \in M$ for a query q is proportional to the Kullback-Leiber divergence between the language models of q and d .

To capture the *interestingness* of a document for a user u , rather than just its general relevance to the query, we incorporate the user profile as another language model. Specifically, we use a mixture model over the estimated language models $\theta_q, \theta_d, \theta_u$ for the query, document and user, respectively. The parameter λ determines the relative weight of the query and user model:

$$\text{score}(q, d, u) = -(\lambda \mathcal{D}(\theta_q \| \theta_d) + (1 - \lambda) \mathcal{D}(\theta_u \| \theta_d)) \quad (3.1)$$

where \mathcal{D} is the Kullback-Leiber divergence, which is calculated as:

$$\mathcal{D}(\theta_x \| \theta_d) = \sum_{w \in V_x} p(w | \theta_x) \log \frac{p(w | \theta_x)}{\frac{f(w, d) + \mu p(w | \theta_C)}{|d| + \mu}} \quad (3.2)$$

where $\theta_x \in \theta_q, \theta_u$ represents either the query language model θ_q or user language model θ_u , V_x denotes the vocabulary of the corresponding language model, $p(w | \theta)$ is the estimated probability of word w in the language model θ , $f(w, d)$ is the frequency of word w in document d , and $|d|$ indicates the length of document d . Dirichlet smoothing is applied using the parameter μ and a background language model θ_C .

Incorporating Word Embeddings. Optionally, we integrate word embeddings to enhance semantic matching; our implementation uses pre-computed word2vec vectors [Mikolov et al., 2013]. The language model is augmented by a translation model, largely following [Kuzi et al., 2016].

$$\begin{aligned} p(w | \theta_d) &= \sum_{t \in V_d} p(w | t) p(t | d) = \\ &= \sum_{t \in V_d} p(w | t) \frac{f(t, d)}{|d|} = \\ &= \frac{1}{|d|} \sum_{t \in V_d} p(w | t) f(t, d) \end{aligned} \quad (3.3)$$

where $p(w | t)$ is the probability of translating term t into w and V_d is document vocabulary.

We calculate this probability using the cosine similarity between word-embedding vectors, normalized to form a probability:

$$p(w|t) = \frac{\cos(w, t)}{\sum_{t' \in V_d} \cos(t', t)} \quad (3.4)$$

Finally, by substituting $f(w, d)$ (Equation 3.2) with $p(w|\theta_d) \cdot |d| = \sum_{t \in V_d} p(w|t) f(t, d)$, we can calculate the Kullback-Leiber divergence as:

$$\mathcal{D}(\theta_x || \theta_d) = \sum_{w \in V_x} p(w|\theta_x) \log \frac{p(w|\theta_x)}{\frac{\sum_{t \in V_d} p(w|t) f(t, d) + \mu p(w|\theta_C)}{|d| + \mu}} \quad (3.5)$$

BM25

To personalize the BM25 [Robertson and Zaragoza, 2009] scoring, we use query expansion techniques, treating the user profile as a bag-of-words. Optionally, instead of expanding the original query with the user profile terms, we can replace the whole query with the user profiles, especially since our focus is on the second-stage re-ranking of a pool of query-relevant candidate results.

Neural Rankers

We adopt two neural ranking methods: DRMM [Guo et al., 2016] and PACCR [Hui et al., 2018]. DRMM considers term-term similarities between query and document terms, utilizing a matching histogram pooling mechanism and a feedforward network to compute a matching score. PACCR also takes the query-document term-term similarity matrix as input, but unlike DRMM, it is position-aware and utilizes CNN kernels and max-pooling to extract matching n-gram features.

Analogous to BM25, we focus on re-ranking and treat the user profile as the query, represented as a term sequence where each term is embedded using word2vec. In contrast to the other ranking methods, the neural rankers require training data. To this end, we use a subset of user-query pairs for training and perform cross-validation.

3.3 Data Collection

3.3.1 Queries and Documents

In this study, we focus on the **book domain** as an exemplary case for search-based recommendation.

Queries. We consider medium-grained query topics: finer-grained than merely specifying a genre of interest (e.g., “history” or “science fiction”), but not as specific as aiming for a single entity as the answer. We gather 50 medium-grained queries by crowdsourcing, examples include:

historical romance, african books, alternate history, dark psychological fiction, greek mythology, historical fiction, memoirs and autobiography, novels made into movies, political science, scandinavian suspense, sword and sorcery, sci fi noir, time travel, victorian society

Documents. The entity-level documents, each representing a book, are collected by running a commercial search engine on the book community website [goodreads.com](https://www.goodreads.com) with the queries as input. The top-100 retrieved results per query form the candidate item pool—each item is a single book. The book pages are scraped; they consist of book metadata (e.g., book title, genre tags, a short description), and user reviews written for the book.

3.3.2 User Study

In order to investigate sparse questionnaire-based user profiling in personalization, and due to lack of such public data, we perform a two-stage user study. We conduct an Amazon MTurk study in which we recruited 33 people for a two-stage task. In the first stage, users create sparse profiles by completing a questionnaire as explained in Section 3.2.2. Our short questionnaire (Figure 3.1) consists of only 10 questions which took an average of 7 minutes to complete.

In the second stage, for search-result evaluation, we asked the same participants to provide judgments of *interestingness* for the results of up to 5 self-selected queries, which they deemed of personal interest. The judgments were graded as follows: “*not interesting*” (0), “*interesting*” (1), “*very interesting*” (2), or “*don’t know*” (discarded). To ensure faithful judgments, we required a justification sentence for each judgment; examples of such justifications are shown in Figure 3.2.

The result pool consisted of 50 queries, each with up to 100 retrieved results. Since it is not feasible for every participant to evaluate all 100 results per query, we reduced the number of queries judged by each participant and the number of results assessed per query. Specifically, we randomly selected 20 results per query for evaluation. To minimize bias toward globally popular items, we avoided selecting the top 20 results from baseline rankings or pooled rankings. In total, we collected judgments for 115 user-query pairs from 33 participants, covering 47 distinct queries and resulting in 2163 judged items.

3.4 Experiments

3.4.1 Research Questions

Our experiments aim to obtain insight on the following research questions.

- RQ1: To what extent can sparse user profiles improve rankings towards individual interests? We compare non-personalized and personalized versions of all ranking methods of Section 3.2.3.
- RQ2: Do entity descriptions improve the ranking? We investigate ranking variants with and without entity models (except for neural methods which cannot handle long text as query input).
- RQ3: Do word embeddings improve the ranking by semantic similarities between terms? We examine this by running the language-model-based ranker with and without embeddings.

3.4.2 Experimental Setup

Metrics. To evaluate rankings, with the graded user judgments, we use normalized Discounted Cumulative Gain (NDCG@5 and NDCG@20) and Precision@1 applied to condensed lists, with all unjudged results filtered out. This follows [Sakai, 2007], which argues that this approach to handling partial judgments is preferable to other metrics. To compute precision, a result was considered good ($= 1$) when deemed “very interesting” or “interesting” by the respective judge.

Entity Linking (Section 2.3). To disambiguate entity mentions in user profiles, we employed the AIDA tool [Hoffart et al., 2011], which links each mention to its corresponding entry in the YAGO knowledge base [Hoffart et al., 2013] and, consequently, to its Wikipedia page. To ensure accuracy, we manually reviewed and corrected a small number of links to eliminate potential errors and reduce the risk of erroneous drift. For entity descriptions, we used the first paragraph of each entity’s Wikipedia article as the source.

Hyperparameters. Due to the limited size of our dataset (despite considerable spending on MTurk), we minimized the tuning of hyper-parameters and picked reasonable defaults where possible. We use the following setup and hyper-parameters for each family of methods:

Language model ranker:

- λ determines the relative influence of query and user models. To test extreme cases, we either set $\lambda = 0$ or $\lambda = 1$.
- μ , the Dirichlet prior parameter for smoothing, was set to the average document length for each query.
- N-gram order in the language model approach is set to 1.
- Language models are estimated using Maximum Likelihood Estimation (MLE).
- When incorporating embedding similarity into the language model, we discarded terms with a similarity below $T = 0.5$.

- θ_C is the background model, which is based on the ClueWeb09 dataset¹.

BM25:

- We set $k_1 = 1.5$ and $b = 0.75$, after determining that a grid search of $[0.1 - 4.0]$ for k and $[0.1 - 1.0]$ for b with stepsize of 0.1 did not improve results.

Neural rankers:

- We use ten-fold cross-validation (with each of the ten folds containing unique queries): eight folds for training and the remaining two for validation and testing.
- The models are trained with a softmax loss over pairs of documents.
- DRMM used an IDF gate with LCH-normalized histograms.
- PACRR used unigrams through trigrams, 32 filters, a k -max of 2, and a combination layer of size 32.
- Due to the substantial increase in input length, we do not consider using entity descriptions with neural models.

3.5 Results

3.5.1 Main Findings

Methods	NDCG@20			NDCG@5			Precision@1		
	Query	User Profile	+ Entities	Query	User Profile	+ Entities	Query	User Profile	+ Entities
LM	0.783	0.797	0.772	0.557	0.576	0.525	0.765	<u>0.817</u>	0.687
LM-embed	0.782	0.781	0.776	0.558	0.548	0.541	0.748	0.722	0.704
BM25	0.781	0.799	0.776	0.555	0.579	0.536	0.774	0.809	0.696
DRMM	0.752	0.782	-	0.493	0.544	-	0.644	0.730	-
PACRR	0.766	0.792	-	0.53	0.572	-	0.67	0.809	-
Commercial SE	0.757	-	-	0.505	-	-	0.652	-	-

Table 3.1: Results for different rankers when using the queries, user profiles, and user profiles enhanced by entity descriptions.

The results are shown in Table 3.1, rows indicate the re-ranking methods (LM denotes the language model method, LM-embed additionally uses word2vec embeddings), and columns show the different inputs to the ranker: query, user profile, and user profiles enhanced with entity description. For each reported metric, the best result for every method across different inputs is highlighted in **bold**, while the overall best result is underlined. As a reference, we also include the performance of the original ranking retrieved from a commercial search engine, denoted as Commercial SE.

Regarding RQ1, comparing personalized vs non-personalized results, we observe that incorporating sparse user profiles consistently improves performance across methods and metrics, with the exception of LM-embed. We performed paired t-test over all

¹<https://lemurproject.org/clueweb09/>

samples of non-personalized vs. personalized rankings; the p-values for the hypothesis that personalization is beneficial were below 0.01 for all metrics, confirming the statistical significance of these improvements.

Regarding RQ2 and RQ3, neither word embeddings nor entity descriptions helped to improve rankings; in fact, their performance often dropped below the non-personalized baselines. We believe that the word2vec embeddings are too broad, and diluting the focus of the inputs. This suggests the potential need for user-specific and domain-specific embeddings, though it remains an open question how such embeddings could be effectively constructed. Additionally, the use of translation models that aggregate semantically similar document terms may introduce noise, further detracting from performance.

The negative impact of entity descriptions is likely due to their broad scope, which includes entities such as locations, movies, books, and general concepts. Analyzing incorrectly high-ranked results revealed that ambiguous or out-of-scope entity descriptions can mislead the ranking. To address this, our model could benefit from considering additional measures, such as entity specificity, to selectively incorporate relevant entities and descriptions while filtering out irrelevant or ambiguous information.

3.5.2 Ablation Study

Methods	NDCG@20			Precision@1		
	Full	Limited	Minimum	Full	Limited	Minimum
LM	0.797	0.776	0.753	0.817	0.748	0.617
LM-embed	0.781	0.766	0.758	0.722	0.704	0.704
BM25	0.799	0.792	0.781	0.809	0.774	0.687
DRMM	0.782	0.774	0.771	0.730	0.661	0.687
PACRR	0.792	0.785	0.783	0.809	0.722	0.713

Table 3.2: Experiments on user profiles without book information (Limited), and with only demographics and hobbies (Minimum) compared against the complete user profiles (Full).

We performed an ablation study with further restriction of user profiles, to investigate how minimal the profile could be while still being beneficial. As shown in Table 3.2, we evaluated two variants: 1) Limited: omitting the book-related fields from questionnaires (but keeping fields on movies and music), and 2) Minimum: keeping solely the demographic attributes and the hobbies field.

The first restriction led to a notable loss in NDCG and precision, but still gave decent quality, whereas the variant with minimal profiles resulted in a significant degradation in performance. For example, when using the BM25 ranker, full profiles yielded 80%

NDCG@20 and 81% Precision@1. Omitting the book fields reduced performance to 79% NDCG@20 and 77% Precision@1, while using the minimal profile further decreased it to 78% NDCG@20 and 69% Precision@1. This suggests that capturing users’ core interests and tastes is crucial. Interestingly, cross-domain knowledge (e.g., using favorite movies to recommend books) may still provide valuable insights.

3.6 Related Work

User Modeling. User modeling is the first step towards personalization; it involves approaches to extract, generate, or learn a user model, ways to store and represent the model, sources and types of data used in profiling the users. Latent and data-intensive user profiling in personalized search and recommendation has been long studied. For example, in personalized search [Agichtein et al., 2006] modeled users based on their browsing behavior such as clickthrough logs and dwell time, and [Teevan et al., 2005] pioneered the analysis of user interests and activities from multiple data sources including query-click logs, email histories, calendar events and even online contents written or read by a user. Similarly, in collaborative-filtering-based recommenders, latent user representations are learned based on user-item interactions [Koren et al., 2009], whereas content-based recommenders use item metadata or user reviews to create user profiles, often comprising long and extensive data.

Ontology-based [Chirita et al., 2005, Stamou and Ntoulas, 2009] and tag-based [Sen et al., 2009, Balog and Kenter, 2019] user profiles are more understandable by users, thus increasing system interpretability. However, they are usually extracted from large data logs, making them as “data-hungry”. User demographic features have also been considered as a source for personalization [Yang et al., 2016], but these are often deemed too crude and uninformative.

[Balog and Kenter, 2019] has formulated a vision and research agenda for constructing *personal knowledge graphs* (PKG) and leveraging them in various personalization and data management tasks. Our questionnaire-based user profiles can be viewed as and easily adapted into a structured PKG, making this work an investigation into such data structures within personalized ranking tasks.

Interview-based preference elicitation methods typically involve presenting an item for the user to rate or showing two items for a pairwise comparison [Sepliarskaia et al., 2018, Rashid et al., 2008, Rashid et al., 2002, Rokach and Kisilevich, 2012, Christakopoulou et al., 2016]. However, these methods differ entirely from our questionnaire-based profiling, which adopts a more natural, human-like approach by acquiring user preferences through textual responses, rather than requiring ratings or comparisons of items that users may not be familiar with.

Personalized Ranking. The second step is to leverage the user model for personalization tasks, such as answer ranking, query expansion, and auto-completion

suggestion [Matthijs and Radlinski, 2011, Shokouhi, 2013, Cai and de Rijke, 2016a]. Our focus is on the personalized ranking task, which has been studied from two perspectives: personalized web search and recommendation.

For *personalized web search*, [Sontag et al., 2012] developed methods for incorporating user-specific priors into language models, [Wang et al., 2013] trained personalized neural ranking models, and [Teevan et al., 2005] adapted a relevance feedback framework. Query expansion and reformulation has also been studied in this setting [Zhou et al., 2017, Yao et al., 2020]. Similarly, *content-based recommenders* mainly perform text matching between the user profile and item metadata, with state-of-the-art methods employing deep learning [Zheng et al., 2017, Suglia et al., 2017, Pugoy and Kao, 2020].

Search-based Recommendation. A similar paradigm to search-based recommendation is *personalized entity search*, where entities can range from people and organization to books, movies and products. For traditional entity search (e.g., people, organizations, locations), prior work on personalization is scarce. However, research on item-level entities, such as entertainment and e-commerce, has been further explored. For example, CLEF had a series of competitions on book recommendations [Koolen et al., 2016], but this relied on posts, tags, reviews and ratings by many users in the LibraryThing community and the Amazon.

The closest to the search-based recommendation paradigm is personalized product search. For instance, [Ai et al., 2017] proposed learning embeddings for users and items within the same semantic space, leveraging user-written reviews on item pages. However, relying on such extensive user data reduces the scrutability and interpretability of the system. Lastly, [Zamani and Croft, 2020] investigated a joint model for search and recommendation tasks, but the two tasks remained separate, and the joint task was not studied.

3.7 Conclusion

In this chapter, we explored questionnaire-based user profiling in the context of search-based recommendation. We constrained the user profile to be concise, understandable, and actionable. To this end, we conducted a user study, collecting a new form of user data for profiling, along with user-provided assessments. We employed a two-stage retrieval and re-ranking process and experimented with various re-ranking approaches for personalization. Our findings indicate that even sparse user profiles can improve ranking quality through personalization. The overriding goal of this work is to understand the boundaries of personalization quality, even with minimal user data.

Chapter 4

Chat-based User Profiles

Contents

4.1	Introduction	48
4.2	Methodology	50
4.2.1	Overview	50
4.2.2	Entity Expansion	51
4.2.3	Domain-specific Vocabulary Weighting	53
4.2.4	Re-ranking Methods	53
4.3	Data Collection	55
4.3.1	Domains, Queries, and Documents	55
4.3.2	User Study	56
4.4	Experiments	58
4.4.1	Research Questions	58
4.4.2	Experimental Setup	59
4.5	Results	61
4.5.1	Main Findings: RQ0 and RQ1	62
4.5.2	Domain Vocabularies: RQ2	63
4.5.3	Entity Expansion: RQ3	64
4.6	Related Work	64
4.7	Conclusion	66

In the previous chapter, we investigated concise questionnaire-based user profiles for search-based recommendation. These questionnaires explicitly capture users' preferences, interests, and general demographics. Our findings showed that even sparse user profiles can enhance personalization performance.

In this chapter, we explore tapping into implicit signals of interests and tastes embedded in users’ online behavior. Specifically, we investigate online chats between two humans as a source for user profiling. Unlike explicit questionnaire-based profiles from Chapter 3, chat-based profiles are richer but less structured. However, they remain more interpretable than clicks and scroll logs, as they are text-based and human-comprehensible.

We compare the personalization effectiveness of these two text-based profiling approaches: concise questionnaires versus extended chat histories. Through extensive experiments across various domains, we find that both approaches significantly improve search-based recommendation over non-personalized baselines, with each method showing advantages in different areas.

4.1 Introduction

Motivation and State of the Art. *User profiling* is a core aspect of personalization, and the increasing availability of online user data provides diverse sources for building user models [Eke et al., 2019]. User models can be represented *explicitly*, such as in textual form or structured graphs, or they can be *implicitly* modeled as latent embeddings. These models can be constructed from various observations of user behavior, ranging from *explicit* signals of preferences to *implicit* cues about their interests.

One of the most widely used signals of user preferences comes from explicit interactions such as likes, ratings, and purchases [Zhao et al., 2019, Lalmas, , Jiang et al., 2020], as well as implicit in-platform behavior such as clicks and dwell times [Liu et al., 2010, Wu et al., 2020], which are primarily used to learn latent user representations. However, this type of data is typically accessible only to service providers and, despite carrying strong signals of interests, remains unintuitive for users to interpret and control.

Another use of the user behavior logs is to create explicit and semi-structured user profiles, such as by identifying topics of interest. In this case, users can interpret and modify their profiles by validating or removing suggested topics [Wu and Grbovic, 2020, Mysore et al., 2023]. A well-known example is adssettings.google.com, where users can partially customize their profiles. However, although offering some degree of user control, these profiles are derived from extensive proprietary logs of users’ browsing behavior and online activities, often incorporating patterns from similar users alongside individual data.

Despite their potential, implicit textual signals from online behaviors—such as social media posts or user chats—remain underexplored in academic research on user profiling, even though they are likely leveraged in industry settings. These signals contain rich information about user interests but introduce challenges due to their

inherent noise and ambiguity. *Social recommendation* encompasses recommendation systems that operate within social media platforms, including those that suggest people, posts, or communities, as well as any system that leverages social relationships or content within these networks [Tang et al., 2013b]. Examples include tourism recommendations [Menk et al., 2019], friend recommendations [Tang et al., 2013a], and social media item (content) recommendations [Guy et al., 2010, Kywe et al., 2012, Sun et al., 2015].

In this chapter, we investigate how online chats between users can be leveraged for search-based recommendation. To the best of our knowledge, this is the first work to explore chats as a source for personalization.

Approach and Contributions. The primary focus of this chapter is leveraging signals from user chats for personalized search-based recommendations across a variety of domains. To this end, we record real-time conversations, gathered in a substantial user study with 14 local participants. The collected chat data consists of 83 pair-wise chats with more than 9k utterances and 59k tokens, and a total duration of 93 hours. To contrast the chat-based profiles to the questionnaire-based ones (Chapter 3), the participants were asked to fill out several questionnaires as well.

Similar to the setup in Chapter 3, for the search-based recommendation, we use a two-stage retrieval and re-ranking paradigm. Where the non-personalized results are obtained using a search engine with the query, and second-stage re-ranking performs personalization. We devise various re-ranking techniques: statistical language modeling (LM), BM25, and neural ranking methods. We also explore entity detection and entity expansion to enrich the user models.

A novel contribution in this chapter, is enhancing the static re-rankers (LM and BM25), that lack the ability to learn additional term importance, with domain-specific vocabulary weighting. On the domain-awareness direction, this chapter also explores selective entity expansion by their similarity to the recommendation domain.

This work makes the following contributions:

- It is the first approach to consider user chats as a source for search-based recommendation across a variety of vertical domains. Chats are a rich source of information about individual interests and tastes. In contrast to latent models learned from clicks, likes, ratings, etc., a user can more easily interpret and edit/censor this information to selectively restrict its usage for privacy reasons.
- We systematically compare chat-based personalization against a more restrictive approach that merely uses concise user profiles based on short questionnaires. In our experiments, both show advantages in certain domains, and perform on par overall.
- We introduce methods for per-domain customization by controlling the vocabulary and appropriate weighting of terms, and demonstrate their effectiveness through experiments.

- We devise techniques to harness entities and background knowledge in the construction of user models with respect to the recommendation domain, and we report on their experimental effectiveness.
- We release a dataset consisting of filled questionnaires, pair-wise user chats, document URLs, and search result assessments by users for three domains (books, travel, food). The data is available at <http://personalization.mpi-inf.mpg.de/>

4.2 Methodology

4.2.1 Overview

We approach the search-based recommendation task by re-ranking an initial pool of non-personalized results, retrieved based on query relevance, using three different scoring methods: statistical language modeling, the BM25 family, and neural ranking. Building on these ranking methods, we incorporate a user model to personalize results and apply domain-specific term weights to identify informative terms within a domain. Additionally, we expand entities found in the user model and explore selective entity expansion based on domain relevance.

Let us define the key components considered across the various approaches discussed in this section:

- **Domains** represent broad recommendation categories, such as books, travel, and food.
- **Queries** are short, medium-grained bags of words (or phrases) that describe a specific context or need the user is currently exploring.
- **Documents** are entity-level documents obtained from specific websites that provide comprehensive content within each domain.
- **User models** represent a user’s interests and preferences as a bag of words (or n-grams) derived from various data sources. In this study, user data originates from either a short *questionnaire/profile* filled out by the user or a *collection of online chats* with other users. Both sources are refined by instructing users to focus on specific topics: general, books, travel, and food. This results in eight base user model variations, which we further enhance by expanding entities within them.

For illustration, Figure 4.1 presents excerpts from the questionnaire and chat collection for an example user. For the query “temples and culture”, this user-specific information led to high rankings for travel destinations such as Borobudur, Delphi, and Ellora—all validated as excellent recommendations by the user. Further details on data collection, including how such data is gathered, are provided in Section 4.3.

<p>- How often and how long do you travel for leisure? Occasionally for short trips (3-5 days)</p> <hr/> <p>- Which countries (or regions) have you ever traveled to? Through the north of Colombia and the south of Germany</p> <hr/> <p>- Which locations (countries, regions, cities, landmarks) and activities did you enjoy the most ? It doesn't matter the country, region or city, I love to go to museums (every museum) historic buildings and monuments. I really love art museums in Bogota, Paris, Rome and Munich</p> <hr/> <p>- Name 3 places that you would like to visit? Tokyo, Disneyland US, Machu Picchu</p> <hr/> <p>- Describe your dream vacation in a sentence or two! Traveling through the most important cities for the art history in the world</p>	<div style="border: 1px solid black; border-radius: 10px; padding: 10px; margin-bottom: 10px;"> <p>Even the Rome colosseum ... Actually I was a bit disappointed about it when I went inside</p> </div> <div style="border: 1px dashed black; border-radius: 10px; padding: 10px; margin-bottom: 10px;"> <p>Ohh, but why? it looks nice in the photos which i have seen</p> </div> <div style="border: 1px solid black; border-radius: 10px; padding: 10px;"> <p>Rome ruins are something that everyone abandoned for a long time and then tried to take care of ... My mother is an architect and I'm an art student so of course Rome monuments have a strong influence in my life</p> </div>
--	---

Figure 4.1: Excerpts from user questionnaire and chat on travel domain (with recognized named entities and concepts in **boldface**).

4.2.2 Entity Expansion

Entity Linking (Recognition and Disambiguation). Among all terms and phrases in the user’s chats and questionnaires, named entities and concept entities deserve specific treatment. To this end, we aim to identify text spans that denote entities (recognition stage), and disambiguate and link them to uniquely identified entities in a knowledge base (disambiguation stage).

For the recognition stage, initially, we employed a standard Named Entity Recognition (NER) tool (stanfordnlp.github.io). However, the automatic NER produced both many false positives and false negatives on the chat data. This is largely caused by the very colloquial nature of user chats, with short-hands, misspellings, ungrammatical utterances and ad-hoc choice between upper-case and lower-case. To mitigate this effect, we hired crowdsourcing workers to mark up text spans for named entities and also general concept entities that exist in Wikipedia (e.g., “history” or “Buddhist art”). This way we eliminated nearly all NER errors.

We then performed Named Entity Disambiguation (NED), disambiguating entity mentions and linking them to their canonical entries in a knowledge base, given the perfect entity mention mark-ups of the NER stage. As a result, the NED stage performed well, with precision reaching approximately 0.83 (estimated by sampling). Noting that we only obtained the manual perfect mark-up for NER stage as this is much easier for crowd workers than NED (e.g., “India” sometimes denoting the Indian Cricket Team rather than the country). We used a standard automatic disambiguation tool (github.com/ambiverse-nlu), linking entities to the YAGO knowledge base and consequently to their Wikipedia pages.

User Model Enrichment. Rather than directly adding the names of the detected entities to the user models, which may lead to overfitting due to dealing with long-tail entities (e.g., lesser-known books or niche travel destinations), we experimented with enriching user models by expanding the entities with additional terms. We explored two approaches for entity expansion: using embeddings and using descriptions.

We first conducted pilot experiments with entity embeddings using Wikipedia2vec [Yamada et al., 2016, Yamada et al., 2018] to achieve proper generalization. However, the results were not satisfactory: many terms that are highly related by Wikipedia2vec turned out to be quite uninformative or even misleading (e.g., “history” being most related to “literature”, or “modern”, “natural” and “wine” being most related to “coffee”, “beer”, or “food”). To avoid this noise and topical drift, we expanded the entities using their descriptions from the first paragraph of their Wikipedia articles. The entity descriptions consist of concise summaries, such as overviews of books, highlights of travel destinations, and similar information.

Selective Expansion by Domain-relevance. Some of the extracted entities may be poor cues for a certain target domain. For example, a user chatting about “Italian cuisine” may not be helpful for book recommendations and could even be misleading for travel destination suggestions. To address this potential dilution, we use domain-entity relevance to filter candidate entities before enriching the user model.

To do this, we construct a domain model for each domain using Wikipedia2vec embeddings, which capture both entity-level relationships and textual descriptions [Yamada et al., 2016, Yamada et al., 2018]. Candidate entities are mapped into the same latent space, and cosine similarity between an entity and a domain is used to select entities above a threshold.

Specifically, the domain representative vectors are computed by aggregating neighboring words and entities of the Wikipedia articles on the domains: “book”, “travel” and “food” for each respective domain. The neighboring words and entities are determined based on cosine similarity between vector embeddings. Similarly, for each entity, we aggregate its neighbors to represent the entity.

Finally, for selective entity expansion in per-domain user models, we select entities whose similarity to the respective domain model is above a specified threshold.

This approach introduces 10 hyperparameters for each domain, including the number of neighbors used to create domain and entity representations, the aggregation functions, and the similarity thresholds for pruning entities. We tuned these hyperparameters via grid search to maximize the area under the precision-recall curves for the domain-entity matching task. Manually annotated entities from the domain-specific questionnaires served as the ground truth for domain relatedness. The complete list of hyperparameters is provided in Section 4.4.

4.2.3 Domain-specific Vocabulary Weighting

Humans have a natural tendency to talk about a diverse set of topics even within the course of a single conversation, for example they may start by talking about their hometowns and cultures, then diverge to their bucket list of travel destinations, and then change to their preferred cuisines, maybe even talking about their allergies. When using the chats as a source for user modeling in a recommendation task, we want to emphasize on domain-relevant parts of the user data. The intuition is that terms in a user chat are informative if they refer to a certain meaning within a particular domain. For example, terms like “history”, “museum” or “nature” are good cues about a user’s travel interests, whereas terms like “price”, “bargain” or “allergen” are uninformative—although all these terms have comparable IDF values in large corpora.

We incorporate this idea of domain specificity by computing per-domain weights for terms, and impose weighted term contributions when ranking documents (Section 4.2.4). To this end, we estimate the conditional probability of a term occurring in a domain-specific context (Dom) given that it occurs in a general corpus (All):

$$\text{spy}_{\text{dom}}(w) = P(w \in Dom | w \in All) \propto \frac{tf(w \in Dom)/|Dom|}{tf(w \in All)/|All|} \quad (4.1)$$

where $tf(w \in C)$ denote the frequency of term w in text collection C .

As the underlying text collections for this estimator, we use the pool of retrieved documents per domain (e.g., book pages with their metadata, such as book descriptions and user reviews) as the Dom text collection. The general corpus, All , consists of text collections from multiple domains.

We also experimented with term weighting for user-specific vocabularies, contrasting all chats by the same user against a universal corpus. This did not lead to significant changes in the empirical results, though, and is disregarded in the following.

In Section 4.2.4, we explain how the ranking models are augmented with the domain-specific term weights.

4.2.4 Re-ranking Methods

Given a query q , a pool of non-personalized results M , and a user model u , we personalize the results by re-ranking them according to the user model. We explore three re-ranking methods for doing this.

Statistical Language Models

The first variant for re-ranking is based on language models (LMs) [Zhai, 2008], which provide a natural way to incorporate the user model. We compute the Kullback-Leibler divergence between a query model and a document model with Dirichlet

smoothing over unigrams or n-grams. To personalize for a specific user, we compute the Kullback-Leibler divergence i) between the query q and the document d and ii) between the user model u and the document d . These two components are combined into a linear mixture with hyper-parameter λ . Additionally, we enhance the ranker with domain awareness by incorporating term weights $\text{spy}_{\text{dom}}(w)$ which reflect the specificity of a term w for a given domain (e.g., books, food or travel), as described in Section 4.2.3. This can be viewed as conditioning the query and user models with a domain model as spy_{dom} is estimation of $p(w \in \text{dom} | p \in \text{All})$.

The score of a query, document, user triplet is calculated as:

$$\begin{aligned} \text{score}(q, d, u) &\propto -(\lambda \text{div}(\theta_q \| \theta_d) + (1 - \lambda) \text{div}(\theta_u \| \theta_d)) \propto \\ & -\lambda \sum_{w \in V_q} \text{spy}_{\text{dom}}(w) \cdot p(w | \theta_q) \log \frac{p(w | \theta_q)}{(p(w | \theta_d) + \mu p(w | \theta_C)) / (|d| + \mu)} \\ & -(1 - \lambda) \sum_{w \in V_u} \text{spy}_{\text{dom}}(w) \cdot p(w | \theta_u) \log \frac{p(w | \theta_u)}{(p(w | \theta_d) + \mu p(w | \theta_C)) / (|d| + \mu)} \end{aligned} \quad (4.2)$$

where V_u and V_q are the vocabularies of the user and query models, and θ_q , θ_u , θ_d and θ_C denote the multinomial parameters of query, user, document and background models, with Dirichlet smoothing parameter μ .

Optionally, we integrate word embeddings by using the cosine distance between pre-computed word2vec embeddings [Mikolov et al., 2013] as a term-term similarity score $\text{sim}(w, t)$. This is plugged into the document model by means of a translation model largely following [Kuzi et al., 2016], with per-term contributions $p(w | \theta_d)$ replaced by summing over all similar terms (above a threshold): $\sum_{t: w \sim t, t \in V_d} \text{sim}(w, t) \cdot p(t | \theta_d)$.

BM25

The second variant for re-ranking is the Okapi BM25 model [Robertson and Zaragoza, 2009]. We incorporate the user model by query expansion. In principle, all terms from the entire chat collection of a user are added to the query. We adapt BM25 with domain-specificity term weights spy_{dom} of Section 4.2.3, dampening the undue term influence of frequently occurring but uninformative (and domain irrelevant) words.

$$\text{score}(q \cup u, d) \propto \sum_{w \in V_{q \cup u}} \text{spy}_{\text{dom}}(w) \cdot \text{idf}(w) \cdot \frac{\text{tf}(w, d) \cdot (k_1 + 1)}{\text{tf}(w, d) + k_1 \cdot \left(1 - b + b \cdot \frac{|d|}{\text{avgdl}}\right)} \quad (4.3)$$

with document length $|d|$, average document length avgdl , and BM25 parameters b and k_1 .

Neural Ranking with KNRM

The third variant for re-ranking is the KNRM neural method [Xiong et al., 2017] which takes a bag-of-words query as input. KNRM produces a query-document relevance score by comparing embedding similarities between query and document terms, leveraging a kernel-pooling mechanism to capture soft-matching signals. A fully connected layer is then used to aggregate these signals into a final ranking score. During training, KNRM learns how to weigh different embedding similarity levels. Similar to BM25, we incorporate the user model by query expansion.

4.3 Data Collection

4.3.1 Domains, Queries, and Documents

Vertical Domains. This work investigates search-based personalization in three domains: *books*, *travel*, *food*. These domains are strategically chosen, since they are more long term and less effected by trends.

Queries. For each domain, we compiled 25 medium-grained keyword queries (total 75 queries). Example queries are shown in Table 4.1.

Books	Travel	Food
Scandinavian suspense	Weekend trip for festival	Perfect breakfast
Novels made into movies	Best wine lover destination	Iron rich vegetarian recipes
Personal development	Epic road trip	15-minute meal recipes
Fairy tale retelling	Cities known for architecture	Brain-boosting recipes
Life in a foreign country	Fitness vacations	Freezable recipes

Table 4.1: Example queries by domain.

Documents. Query results are entity-level documents (or items) obtained from specific websites that provide comprehensive contents about three domains using a commercial search engine with site restrictions: goodreads.com for books, wikivoyage.org for travel, and allrecipes.com for food. The top-100 answers were retrieved, keeping only those that were about specific entities and discarding general list pages-this left us with 90 or more answers for each query. Each answer has a key item that can be easily identified (e.g., from the URL string or page title) and comprises an informative description of the item. Two of the sites (books and food) also include extensive reviews and discussion by their communities.

4.3.2 User Study

We conducted a multi-stage longitudinal user study to obtain user data in various forms: filled out questionnaires, user-to-user chats, and result assessments. Due to the intensity of the study, we opted for a local participation recruitment among university students, recruiting 14 students who were paid ca. 10 Euros per hour.

First, we invited the interested students to participate in a briefing session, in which details of user study and its implications were explained to them. Including details on the tasks, data collection, their right to withdraw from the study at any time, and process of anonymizing and sanitizing their data under their control before releasing. Out of initial 20 students, 16 of whom agreed to participate in the study and 14 of them remained in the full study including the assessment and data publishing.

The first stage consisted of personal data gathering. The most time-intensive part was collecting chat data, where we gathered user-to-user chats over four weeks, with each user participating in three chats per week. For each chat, two users were randomly paired. For the first week, users were instructed to chat generally, like mutual introductions. During the remaining weeks users were asked to chat about specific topical domains: users’ interests and tastes in books and their experience and interests in traveling and food. We recorded a total of 83 real-time user-to-user chats with 15 participants (14 of whom remained until the evaluation phase of the study). The recorded time for chats are over 93 hours, with 9,797 utterances and 59k tokens in total. On average, each user had 2.8 sessions for each domain, totaling to ca. 11 sessions overall, with an average of 653 utterances and 3934 tokens per user. Table 4.2 shows the complete per user statistics of the collected chat data. Example excerpts of user-to-user chats for all four topics are shown in Figure 4.2.

Topic	#sessions	#utterances			#tokens		
	avg.	avg.	min	max	avg.	min	max
General	2.8	181	80	281	983	719	1391
Books	2.7	137	81	209	920	356	1628
Food	2.8	155	85	430	976	435	1636
Travel	2.8	181	81	375	1055	499	2060

Table 4.2: Per-user statistics for the chat data.

In addition, each user completed several questionnaires upfront: a general one with 18 questions about demographics, general interests and personality, and one for each of the themes books, travel and food with 2, 5 and 10 questions. The general questionnaire included personality-oriented questions such as “What are your hobbies?”, “What makes you happy?”, and “Your golden rule?”. Figure 4.3 shows complete questionnaires.

In the last stage, the same users participated in an assessment study of personalized

search results. The users were asked to identify around 5 queries for each domain on themes that looked potentially appealing to them. This way we avoided personalized judgements on areas that the user does not care about. For each query, a user assessed 20 results that were sampled uniformly at random (to avoid ranking bias) and, additionally, the top-10 results from the original non-personalized ranking (with the risk of bias). We asked participants to provide subjective, graded assessments with the following labels: 2 = “strongly interested”, 1 = “mildly interested”, 0 = “uninterested”, and discard = “I don’t know”. Additionally, users were required to provide justification sentences along with their judgments. In total, we gathered 2,673 individual assessments for 113 user-query pairs, covering 73 distinct queries.



Figure 4.2: Example excerpts of collected chat data in different topics.

General Questionnaire	Travel Preferences Questionnaire
<p>Demographics</p> <ul style="list-style-type: none"> - What is your age? - What is your gender? - What is your family status? - Which country are you originally from? - What is your field of education? - what is your occupation? <p>General Information</p> <ul style="list-style-type: none"> - What are your hobbies? - Which news categories do you read? - Which news sources do you read? <p>About You</p> <ul style="list-style-type: none"> - Describe a perfect day. - What do you usually do on a typical weekend? - What was the best day of your life so far? - What makes you happy? - I'm really good at ... - What is your golden rule? - What are your current goals? - What is your dream job? - One day, I would like to ... 	<ul style="list-style-type: none"> - How often and how long do you travel for leisure? - Which countries (or regions) have you ever traveled to? - Which locations (countries, regions, cities, landmarks) and activities did you enjoy the most ? - Name 3 places that you would like to visit? - Describe your dream vacation in a sentence or two!
	<p>Food Preferences Questionnaire</p> <ul style="list-style-type: none"> - What are your favorite meals/foods? Explain why in a short sentence for each. - What is your regular cuisine? - What do you enjoy drinking the most and why? - What are your regular drinks? - What is your diet? - World Cuisine: - Diet and Health: - Dish Type - Cooking Style - Ingredient
	<p>Book Preferences Questionnaire</p> <ul style="list-style-type: none"> - Please name 3 of your favorite books. And explain why you love them in a sentence for each. - Which Genres of books do you like?

Figure 4.3: Complete questionnaires used for data gathering.

4.4 Experiments

4.4.1 Research Questions

Our experiments aim to address the following research questions.

RQ0: Can implicit signals of user interest derived from *user-user chats* be leveraged to personalize search-based recommendations across multiple *domains* including books, food recipes, and travel destinations?

RQ1: How do approaches that utilize individual *conversations* compare to those relying solely on *concise user profiles*?

RQ2: To what extent does *domain-specific customization* improve re-ranking techniques? For example, how do models perform when tailored for different domains, such as books versus travel?

RQ3: To what extent does *entity awareness* add value? Additionally, how important is *domain-related selective entity expansion*? Specifically, how does detecting

entities in user chats, mapping them to a background knowledge base, and (selectively) incorporating this information into user models impact recommendation effectiveness?

4.4.2 Experimental Setup

Evaluation Metrics. As explained in Section 4.3.2, to reduce evaluation bias towards a specific approach, we randomly select 20 items per query from the pool of 100 query-relevant retrieved results and collect user judgments for those. To evaluate our methods on the collected assessments, we primarily report normalized Discounted Cumulative Gain (NDCG@20). We also report Precision@1 by binarizing the user-graded assessments, treating 1 and 2 as positive and 0 as negative. While NDCG@20 evaluates the quality of the entire ranking list, Precision@1 places more emphasis on the top-ranked results.

Additionally, to ensure the quality of the results, we collect user assessments for the top-10 results from the potentially biased rankings provided by a commercial search engine, per query. For this, we report NDCG@10 (denoted as NDCG@top10) to evaluate the top-10 results.

All evaluations are performed on condensed lists, following [Sakai, 2007], by discarding all unjudged results.

Selective Entity Expansion by Domain-relevance (Section 4.2.2). To determine the best domain-entity relevance model, we tuned the hyperparameters through grid search, aiming to maximize the precision-recall curve for the domain-entity matching task using the ground truth provided by the manually annotated entities in the domain-specific questionnaires. We computed precision-recall curves for all possible hyperparameter combinations (excluding the threshold), selecting the ones with the largest area under the curve. The threshold was then chosen based on the point corresponding to the highest precision-recall value.

We tuned 10 parameters per domain, separately for named and concept entities (5 for each).

- Domain/entity num. neigh. refers to the number of neighbors (terms and entities) used to create the domain/entity representation, with options: [0, 5, 10, 25, 50, 100].
- Domain/entity aggregation defines the aggregation method for creating the domain/entity representation, with options: average (avg) and weighted average (wavg).
- Relevance threshold is the cutoff for selecting entities based on their similarity to the domain.

Table 4.3 lists the final hyperparameter values.

Type	Parameter	Book	Travel	Food
Named Entities	Domain num. neigh.	100	25	10
	Domain aggregation	avg	avg	avg
	Entity num. neigh.	100	10	0
	Entity aggregation	wavg	avg	-
	Relevance threshold	0.5725	0.3569	0.2942
Concepts	Domain num. neigh.	5	100	25
	Domain aggregation	avg	avg	wavg
	Entity num. neigh.	100	0	50
	Entity aggregation	avg	-	avg
	Relevance threshold	0.5061	0.5731	0.3874

Table 4.3: Hyperparameters for selective entity expansion.

Methods under Comparison. We cover the following methods and configurations.

- **LM** denotes the language model approach.
 - In pilot experiments, unigrams outperformed bigrams and trigrams; hence the experiments focus on the unigrams.
 - To isolate the effect of the user model in the re-ranking, and as our initial pool of entities are to some extent relevant to the query, we either set the λ to 0 or 1. When $\lambda = 1$ the input to the re-ranker is the query model and when $\lambda = 0$ only the user model is given as input.
 - Dirichlet smoothing is used with parameter μ set to the average document length and a background model θ_C based on ClueWeb’09 ¹.
- **LM-embed** is the language model method with word2vec word embeddings.
 - The term-term similarity threshold is set to 0.5.
- **BM25** is the BM25 method.
 - BM25 parameters are set as $b = 0.75, k_1 = 1.5$.
- **KNRM** is the neural ranker.
 - Maximum input query length is set to 50. For the user model comprise of longer text, the top-50 words are obtained by their term frequency order.
 - Maximum document length is set to 5000 terms.
 - We report on ten-fold cross-validation with 8, 1 and 1 folds for training, vali-

¹<https://lemurproject.org/clueweb09/>

ation and test, respectively. Number of assessments per domain are 504, 772 and 806 for book, food and travel, respectively.

- **KNRM-all** is a single model trained on all three domain data combined. Since the per domain training data is fairly low-end, we combined them into a single training set with 2082 labeled samples and trained the cross-domain model.
- **SE** is the initial ranking from a commercial search engine.

4.5 Results

Ranker	Query Only	User Model							
		Questionnaires				Chats			
		All	Gen	Dom	Dom+Gen	All	Gen	Dom	Dom+Gen
Overall									
LM	0.796	0.816	0.804	0.823*	0.824*	0.811	0.806	0.822*	0.817*
LM-embed	0.794	0.791	0.787	0.811	0.798	0.782	0.777	0.795	0.784
BM25	0.785	0.823*	0.815*	0.827*	0.833*	0.819*	0.816*	0.827*	0.821*
KNRM	0.807	0.791	0.805	0.798	0.794	0.780	0.786	0.784	0.785
KNRM-all	0.810	0.807	0.796	-	-	0.788	0.791	-	-
SE	0.786	-	-	-	-	-	-	-	-
Books									
LM	0.825	0.829	0.823	0.822	0.834	0.846	0.854	0.844	0.847
LM-embed	0.818	0.795	0.803	0.801	0.799	0.811	0.799	0.813	0.806
BM25	0.814	0.843	0.846	0.834	0.847	0.846	0.849	0.851	0.850
KNRM	0.826	0.827	0.832	0.817	0.816	0.790	0.810	0.790	0.809
SE	0.777	-	-	-	-	-	-	-	-
Travel									
LM	0.818	0.821	0.815	0.854*	0.838	0.826	0.813	0.841	0.835
LM-embed	0.813	0.799	0.787	0.849*	0.814	0.785	0.782*	0.803	0.796
BM25	0.794	0.837*	0.833*	0.857*	0.849*	0.836*	0.837*	0.844*	0.838*
KNRM	0.838	0.806	0.833	0.827	0.801	0.800	0.800	0.805	0.800
SE	0.794	-	-	-	-	-	-	-	-
Food									
LM	0.753	0.802*	0.779	0.790	0.803*	0.772	0.766	0.785	0.777
LM-embed	0.757	0.778	0.775	0.777	0.780	0.758	0.755	0.773	0.757
BM25	0.756	0.793	0.775	0.791	0.806*	0.783	0.770	0.793*	0.782
KNRM	0.761	0.751	0.756	0.755	0.771	0.752	0.753	0.757	0.753
SE	0.785	-	-	-	-	-	-	-	-

Table 4.4: NDCG@20 for different rankers and user models. Best results per row are in boldface. Statistically significant improvements over the Query-Only baselines are marked with an asterisk.

4.5.1 Main Findings: RQ0 and RQ1

Table 4.4 shows the NDCG@20 results for the influence of different user models. The top part of Table 4.4 shows the overall results across all domains (averaged over the 113 user-query pairs). The other parts show per-domain results. The user models under comparison here are query-only vs. questionnaires-based vs. chats-based. For the latter two, we varied the specific setting (see Subsection 4.3.2): (*All*) refers to driving user models from all available inputs from all topics, (*Gen*) uses only the general questionnaires or chats, (*Dom*) leverages only domain-specific inputs, (*Dom + Gen*) uses both the general and per-domain inputs. In this comparison, all methods were configured without entity expansion and without domain-specific vocabularies (which will be discussed in the next subsections).

Overall results (top part of Table 4.4): The overriding observation is that almost all rankers with different degrees of personalization improve over the SE baseline and that both questionnaire-based and chat-based user models achieve notable gains over the query-only rankings: in the order of 2 to 4 percentage points in NDCG@20. While the effect size of personalization is only moderate, the relative gains are statistically significant and come at little cost for the ranker efficiency. For significance, two-tailed paired t-tests in comparison to the Query-Only baselines mostly had p-values < 0.05 . These results are marked with an asterisk in Table 4.4. For example, the BM25 gain from 0.785 to 0.833 between query-only and questionnaire-*Dom + Gen* has a p-value of 0.0002.

Interestingly, LM-embed did not improve over LM. The term-term relatedness by word2vec seems to be too crude for our task and dilutes the query focus. Also, we observe the best performing LM-embed variant is the questionnaire-*Dom* where the profile is sparse with only domain specific questions. KNRM and KNRM-all were inferior to the Query-Only case. The combination of small training data and limited input size is the likely cause for this disappointing result.

When comparing questionnaire-based vs. chat-based personalization, the former performs slightly better than the latter, but the differences are minor. For both, the best variants were the ones with user models *Dom* or *Dom + Gen*, indicating awareness of the domain is beneficial. *Dom* is almost always preferable to *Dom + Gen* in the case of chats, but there is no clear trend when using questionnaires. This is likely due to the fact that the general questionnaires were designed to reveal user personalities, whereas the general chats were mostly introductory and less informative. These gains are not always statistically significant, but the best cases are: for example, the improvement for LM from 0.811 with chats-*All* to 0.822 with chats-*Dom* had a p-value of 0.0018.

Per-domain results: The results vary among the different domains in an interesting way. We base the discussion on LM and BM25 as they achieved the best results. For books and travel, the gains from user models are most pronounced. For books, the

chat-based models achieved a small but notable and significant improvement over the questionnaire-based ones. We observe that for questionnaire-based models *Dom+Gen* outperforms *Dom*. This is due to the low coverage of the book domain with only two questions briefly covering the user’s favorite books and genres, whereas the general questionnaire further inquires about demographics and personal traits. On the other hand, for the travel domain with 5 specific questions, *Dom* performs better than *Dom + Gen* in both questionnaire-based and chat-based models, with the former giving the best results.

For food, personalization led to gains, but the absolute NDCG scores were substantially lower than for the other two domains. Here, the SE performed better than the re-rankers with the query-only model. However, using *Dom + Gen* questionnaire-based profiles, we achieved up to 2% improvement over the SE results. It seems that the food domain is inherently difficult to understand, as its vocabulary mixes specific and very common words with a strong influence of the latter on tastes and sentiments (e.g., “hot”, “terrific” etc.).

As for Precision@1, the overall gains by personalization were nearly 10 percent: considering the best-performing rankers on overall results, the LM improved from 70% with query-only models to 81% with questionnaire-based models, and BM25 went up from 66% to 83%. Again, the gains were most substantial for books and travel, but here food as well showed notably improved Precision@1.

We further evaluated NDCG@top10 returned by the SE baseline: not surprisingly, the SE baseline was stronger for this metric, but was still outperformed by re-ranking with personalization. The best values for our method were comparable to those for NDCG@20, around 83% across all domains and up to 87% for travel.

4.5.2 Domain Vocabularies: RQ2

Recall from Section 4.2.3 that we optionally incorporate domain-specific term weighting to reduce the influence of irrelevant terms from the user chats. Table 4.5 shows NDCG@20 results with this awareness of domain vocabularies, for the four chat-based configurations *All*, *Gen*, *Dom* and *Dom+Gen*. We show only overall results across all domains, but for each domain, all user-model terms were weighted by the respective $spy_{dom}(w)$ domain model. For brevity, we restrict ourselves to the LM-based ranker; the findings were similar for the other two rankers.

Domain Specificity	Chats			
	All	Gen	Dom	Dom+Gen
Disabled	0.811	0.806	0.822	0.817
Enabled	0.821	0.813	0.83	0.826

Table 4.5: NDCG@20 for LM-based ranker with domain-specific vocabularies.

Table 4.5 indicates that there are small gains from this domain-specific weighting, but the effect size is marginal and not statistically significant (p-value > 0.1). Among different domains, the travel domain benefited the most from the domain-specific vocabulary weighting. It seems that chats are not sufficiently focused on domain-specific topics. Humans do jump between topics, so chats naturally have a high level of thematic diversity.

4.5.3 Entity Expansion: RQ3

To study the influence of entity expansion for the user models, we compared different settings against the previously reported configurations without entity awareness (*None*). (*All*) expands all entities including both named entities and concepts (in Wikipedia, such as “history” or “Buddhist art”); (*Domain*) restricts the entities to those that are related to the respective domain (see Section 4.2.2); (*NE-all*) uses only named entities (i.e., discarding concepts); (*NE-domain*) uses only named entities with domain relatedness above a threshold.

Entity Expansion	User Models							
	Questionnaires				Chats			
	All	Gen	Dom	Dom+Gen	All	Gen	Dom	Dom+Gen
None	0.816	0.804	0.823	0.824	0.811	0.806	0.822	0.817
All	0.817	0.816	0.826	0.822	0.821	0.814	0.828	0.824
Domain	0.823	0.812	0.827	0.824	0.821*	0.814*	0.829	0.824
NE-all	0.823	0.81	0.826	0.829	0.818*	0.813	0.829	0.825*
NE-domain	0.829*	0.809	0.825	0.833	0.819*	0.815*	0.83	0.824*

Table 4.6: NDCG@20 for LM-based ranker with entity expansion. Best results per column are in boldface. Statistically significant improvements over None baselines are marked with an asterisk.

Table 4.6 shows the overall NDCG@20 for these settings with the different configurations for the user-model construction. We observe that, except for *NE-domain*, none of the expansion methods significantly improve the models derived from questionnaires. The reason is that these models are already very concise given their high-quality inputs. For chat-based user models, on the other hand, entity expansion led to small, but notable and statistically significant (p-values < 0.05), improvements of ca. 1%.

4.6 Related Work

The related work in this chapter focuses on data types and sources for user profiling and does not delve into specific techniques for user modeling or personalization, both of which are covered in previous chapters (Chapter 2 and Chapter 3).

Explicit Preferences. A primary source of explicit user preferences is the record of users' likes and ratings of items within the recommendation platform (e.g., [Sarwar et al., 2001, Zheng et al., 2016]). However, such explicit preference signals are often sparse, making them less desirable as a sole basis for personalization.

Additionally, semi-explicit behaviors, including purchases and online content consumption (e.g., reading, watching, or listening), have also been leveraged for personalization [Shani et al., 2005, Zhao et al., 2019, Steck et al., 2021, Nadai et al., 2024]. While these behaviors provide strong indicators of user interest, they do not express it as explicitly as ratings and have sometimes been considered implicit signals of interest in academic research.

Another source of explicit user preferences is user-provided data through the direct expression of their interests. An example of this approach is our questionnaire-based profiling discussed in Chapter 3. Similarly, [Sanner et al., 2023] conducted a user study where participants provided less structured free-form descriptive profiles, which were then evaluated for their effectiveness in movie recommendation. These approaches have the drawback of requiring user effort and time, making them less practical since not all users are willing to *actively* provide their preferences. Additionally, they need periodic updates to stay accurate, further increasing the burden on users.

Implicit Preferences. A significant portion of research on personalization relies on in-platform implicit preference feedback due to its abundance compared to explicit signals. These include clicks, dwell time, and scrolling behaviors [Hu et al., 2008, Chen et al., 2022].

The overall focus of this thesis is on textual user profiles, therefore the most relevant sources of implicit signals for personalization are user-generated texts, such as social media posts, chats, and reviews. User-written reviews are the most studied source of user-generated text for personalization [Chen et al., 2015, Bauman et al., 2017]. Some studies [Stratigi et al., 2019, Sachdeva and McAuley, 2020] indicate that user reviews contain considerable noise and provide limited benefits. We investigate their role in user profiling in Chapter 5.

Inferring user attributes from social media posts and conversations has been explored by [Tigunova et al., 2020]. However, academic research on using such data sources for personalization remains scarce. It is important to note that interactive and *conversational recommenders* [Schnabel et al., 2020, Lei et al., 2020] are a very different approach, focusing on *session-based* recommendations that build on dialogues between the user and the system.

Another relevant research direction is *cross-domain recommendation* [Zang et al., 2023], which aims to improve recommendation performance in a new domain or application by leveraging data from other domains. It mitigates data sparsity by aggregating user preferences across sources, but aligning interests remains challenging, as preferences may not transfer well across domains.

4.7 Conclusion

This chapter explored user-to-user conversations as a novel data source for search-based recommendations. We employed a suite of re-ranking methods, incorporating enhancements such as domain awareness and entity expansion. Through extensive data collection and experimentation, we demonstrated that both chat-based and questionnaire-based user models significantly improve upon the original search engine results, as well as non-personalized, query-only rankings.

Between chat-based and questionnaire-based approaches, there is no clear winner. Each user modeling paradigm offers specific benefits. Questionnaires are transparent and scrutable; however, they require explicit effort. While most users accept a one-time questionnaire, few are willing to update it periodically as their interests evolve. Chats, on the other hand, require no effort from users and can be updated automatically. However, the derived models are less transparent and not easily adjustable by users. Additionally, chat data poses higher privacy risks.

A promising direction for future work is to enhance the transparency and scrutability of chat-based profiles, for example, by extracting entities from conversations and structuring them into an explicit user model, or by creating a summary from user chats that captures the gist of user preferences in a concise format.

Chapter 5

Concise User Profiles from Review Texts

Contents

5.1	Introduction	68
5.2	Methodology	70
5.2.1	User Profile Construction Approaches	71
5.2.2	Ranking Model	75
5.2.3	Negative Training Samples from Unlabeled Data	77
5.3	Experiments	79
5.3.1	Rationale	79
5.3.2	Datasets	79
5.3.3	Experimental Setup	81
5.4	Results	83
5.4.1	Comparison of CUP against Baselines	83
5.4.2	Comparison of CUP Configurations	86
5.4.3	Efficiency of CUP	88
5.4.4	Influence of Interaction Density	89
5.5	Analysis of User Profiles	91
5.6	Related Work	92
5.7	Conclusion	94

In previous chapters, we explored sparse questionnaires and rich chats as sources for personalization. While the former offers benefits such as *scrutability* and *understandability*, it requires user effort and manual periodic updates. On the other hand, the

latter can be obtained automatically but reduces *user control* due to its length and implicit nature.

In this chapter, we aim to combine the best of both worlds by automatically creating a concise and scrutable profile from long user texts. This allows users to *inspect* and *edit* their profiles at will, without being obligated to do so. This approach improves *transparency* as well as *efficiency* due to the lower computational cost of short profiles as opposed to long inputs.

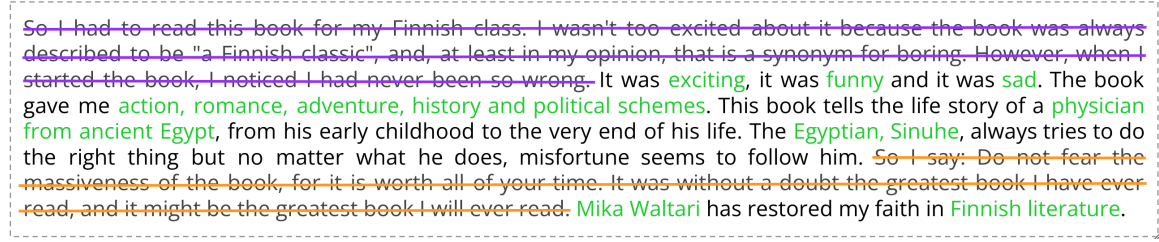
We investigate user-written reviews as a source for profile creation. Review texts are inherently complex, containing informative cues about user preferences as well as noisy signals that may dilute user preference modeling. To address this, we devise various techniques to extract the most informative cues. Our experimental results indicate that judiciously selecting relevant text snippets leads to better performance than using the entire user text.

5.1 Introduction

Motivation. Recommender-system methods fall into two major families or hybrid combinations: i) *interaction-based* recommenders that leverage numeric ratings or binary signals (e.g., membership in personal playlists or libraries) for user-item pairs, and ii) *content-based* recommenders that exploit item features and user-provided content, ranging from metadata attributes (e.g., item categories) to review texts.

In settings where interaction data is *sparse* and *long-tail* items and users are prevalent, content-based methods become the only viable option. The most promising approach in this scenario is to harness *user review texts*, as they offer diverse insights into user preferences. Specifically, in domains where users spend substantial time per item (e.g., books, travel destinations), unlike short-attention-span domains (e.g., music, video streams), users tend to leave detailed reviews that reflect their interests and tastes, even with few interactions.

However, user reviews often contain a mix of informative and uninformative signals, including descriptive elements, sentiment expressions, and personal background. Only a subset of this information is useful for content-based profiling. Figure 5.1 illustrates a review that includes both relevant and irrelevant information. Constructing *concise user profiles* from rich but noisy reviews not only enhances accuracy by filtering out irrelevant content but also increases *scrutability* and *user control* over their profiles while reducing the *computational cost* of processing large text inputs for downstream personalization tasks.



~~So I had to read this book for my Finnish class. I wasn't too excited about it because the book was always described to be "a Finnish classic", and, at least in my opinion, that is a synonym for boring. However, when I started the book, I noticed I had never been so wrong. It was exciting, it was funny and it was sad. The book gave me action, romance, adventure, history and political schemes. This book tells the life story of a physician from ancient Egypt, from his early childhood to the very end of his life. The Egyptian, Sinuhe, always tries to do the right thing but no matter what he does, misfortune seems to follow him. So I say: Do not fear the massiveness of the book, for it is worth all of your time. It was without a doubt the greatest book I have ever read, and it might be the greatest book I will ever read. Mika Waltari has restored my faith in Finnish literature.~~

Figure 5.1: User-written review, with uninformative text crossed over. Personal background is in purple, pure sentiment in orange, most informative cues in green.

In this chapter, we investigate approaches to tackling review-based recommendation with sparse data, long-tail users and items, and rich review texts, with particular attention to system scrutability and computational cost by constraining the size of user profiles.

State of the Art and its Limitations. Recent works integrated item descriptions and textual reviews into various kinds of recommender architectures, including some based on large language models (LLMs) (e.g., [Dong et al., 2025, Huang et al., 2024, Lin et al., 2024, Ramos et al., 2024]). Our approach differs fundamentally, by making user profiles explicit and transparent, before feeding them into a recommender. This way, lay users can inspect, edit, extend or customize their profiles in a human-friendly manner, while personalizing the downstream application.

There are established works on cold-start support and long-tail items (e.g., [Cao et al., 2020, Li et al., 2019, Liu et al., 2022, Luo et al., 2023, Raziperchikolaei et al., 2021]). These are driven by similarity-aware architectures such as graph models, matrix factorization, and neural methods. Their key asset is to infer explicit or implicit properties of long-tail items and the resulting user preferences, by learning from similar items with richer data. This approach does not carry over to long-tail *users*, though, when most users have sparse data and high diversity in tastes and interests.

Approach and Contributions. This chapter focuses on leveraging user-written reviews for personalization, with the desideratum that profiles derived from reviews should be *concise* and *scrutable*, allowing users to inspect, edit, and extend their profiles at any time. Moreover, user profiles should capture *why* a user liked an item based on written reviews while ignoring irrelevant aspects, such as pure sentiment statements or unrelated personal information.

To this end, we propose a lightweight framework, called **CUP**, for constructing Concise User Profiles and utilizing them in a downstream recommender system. The main components of our framework include profile constructors, recommender models, and negative sampling strategies for training and evaluation. This architecture is relatively simple yet highly versatile, supporting different configurations and seamlessly

incorporating a wide range of user profiling techniques.

In the profile construction stage, we extract or generate user profiles from review texts using various techniques, including classical extractive methods (e.g., TF-IDF), generative models (e.g., ChatGPT), and reinforcement learning approaches where recommendation quality serves as a reward signal.

The constructed user profile, along with item metadata, is then used to create a personalized ranking of items. Specifically, we adopt a two-tower transformer-based architecture that enables end-to-end learning of item and user encodings while leveraging a lightweight (small) language model (LM). Alternative architectures, such as cross-encoders, can be easily integrated.

In the absence of explicit negative feedback (i.e., items a user dislikes), the default approach is to sample negative items uniformly at random from unlabeled data. In this work, we explore two alternative negative sampling strategies. The first is genre-based sampling, where negative items share a genre with the positive item. The second is weighted sampling, where item-item similarities are used to assign partial negative and positive weights to the sampled items.

Lastly, in addition to the default evaluation model (uniformly at random sampling), we assess the performance of the models in the *search-based recommendation* mode by simulating user search, where a pool of items similar to the positive test item is retrieved, making the task more difficult.

This study focuses on the domain of books as a prime case for low interaction rates (i.e., users with few items) with high interaction efforts (i.e., value in user reviews), in combination with high diversity of user tastes—both across users and also per user. We conduct extensive experiments, comparing against various baselines, including LLM-based ranking, and thoroughly evaluating different configurations of our framework. Beyond aggregated performance metrics, we also report results broken down by user and item groups.

The key contributions of this chapter are:

- We develop a new framework, **CUP**, for transformer-based recommendation leveraging concise user profiles from reviews.
- We introduce effective techniques for selecting informative cues from long and noisy reviews.
- We conduct comprehensive experiments on data-poor but text-rich users with highly diverse preferences.

5.2 Methodology

In this section, we address personalized (search-based) recommendation by introducing our framework, CUP (**C**oncise **U**ser **P**rofiles). The first subsection explores

methods for constructing user profiles from long review texts. The second subsection describes our transformer-based ranking architecture for ranking items based on user profiles and explains how we simulate search-based recommendation at inference time. The third subsection discusses the challenges of training on unlabeled data and presents alternative negative sampling strategies.

5.2.1 User Profile Construction Approaches

The simplest approach to constructing user profiles from review texts is to concatenate all available reviews into a long token sequence. However, this poses three problems.

First, user reviews contain a noisy mix of aspects, such as:

- preferences related to the reviewed item (e.g., “comedy with a plot twist”),
- long-term interests (e.g., “historical fiction never disappoints”),
- personal background (e.g., “I’m a CS student”),
- general emotions or personal statements (e.g., “I love reading at night”), and
- generic sentiment expressions (e.g., “what a great story”).

Only the first two categories are useful for user profiling, while the latter ones introduce noise and dilute the profile.

Second, the entirety of user-provided text is often too long to be fully processed by a transformer. Even when it fits within the token budget, computational and energy costs scale quadratically with the number of input tokens.

Third, although text-based profiles are user-comprehensible, their excessive length reduces user control over their profile.

To address these challenges, we strictly limit each user’s text profile to 128 tokens and develop a suite of techniques to judiciously select the most informative pieces. Our techniques for selecting the most informative parts of user reviews fall into three categories: extractive methods, zero-shot and few-shot generative models, and trained extractor models.

Extractive Methods

TF-IDF Weighted Phrases. We extract key n-grams from a user’s reviews based on their TF-IDF (Term Frequency-Inverse Document Frequency) scores, prioritizing those with the highest values. The TF-IDF weight for an n-gram t in a user’s reviews is computed as:

$$\text{score}(t) = \text{TF-IDF}(t) = \text{tf}(t, R_u) \cdot \text{idf}(t)$$

where $\text{tf}(t, R_u)$ is the term frequency of phrase t in the user’s entire review set R_u , and $\text{idf}(t)$ is the inverse document frequency of phrase t , precomputed from the Google

Books N-grams dataset to ensure broad coverage of word informativeness, as:

$$\text{idf}(t) = \log \frac{N}{\text{df}(t)}$$

where N is the total number of documents in the corpus, and $\text{df}(t)$ is the number of documents containing t .

N-grams with the highest TF-IDF scores are selected to represent the user profile.

IDF Weighted Sentences. We extract representative sentences from the user’s reviews based on their informativeness, measured using IDF scores. Each sentence s is assigned a weight computed as:

$$\text{score}(s) = \frac{\sum_{w \in s} \text{idf}(w)}{|s|}$$

where $\sum_{w \in s} \text{idf}(w)$ is the sum of IDF values of all words in the sentence, and $|s|$ is the total number of words in the sentence. To prevent longer sentences from dominating the selection, we normalize the scores by sentence length. The IDF values are derived from the Google Books N-grams dataset.

Sentences with the highest IDF scores are selected to form the user profile.

SBERT Scored Sentences. To identify sentences most relevant to the user’s preferences, we compute similarity scores using Sentence-BERT (SBERT) [Reimers and Gurevych, 2019]. Each sentence s from a single user’s review is compared to the description of the corresponding item d using the dot product:

$$\text{score}(s, d) = \langle \mathbf{SBERT}(s), \mathbf{SBERT}(d) \rangle$$

where $\mathbf{SBERT}(x)$ represents the SBERT embedding of text x , and \langle, \rangle denotes the dot product.

Sentences from each review are ranked by similarity scores in descending order. To ensure diversity in the selected set, we employ a round-robin selection strategy, iterating through different reviews rather than extracting all top-ranked sentences from a single review. This prevents overrepresentation of a particular review and enhances the coverage of user preferences.

Generative Models

T5-generated Keywords. We employ a T5 model [Raffel et al., 2020] fine-tuned for keyword generation to convert each review into a set of keywords. These keywords are then concatenated to form the user profile.

ChatGPT-generated Profiles. We feed all of a user’s reviews, in large chunks, into ChatGPT [OpenAI, 2024] and instruct it to generate a keyphrase-based profile

that characterizes the user’s interests. The following prompt is used to guide the model:

```
Here are the user’s reviews of the books the user read:
[REVIEW TEXTS].
Generate some key phrases which can be used to characterize
the type of book content described.
Generate concise list of short key phrases in max 50 tokens
total, print the comma-separated list in single line.
```

Llama-generated Profiles. We instruct Llama [Dubey et al., 2024] to generate user profiles from reviews in two formats: keyword-based and first-person narrative. The model is guided using in-context examples. Below is an example prompt that includes a single in-context example, followed by the user’s reviews, guiding Llama to generate an abstractive profile in the first-person narrative style. Additional in-context examples can be added in a similar manner.

```
You are a book wizard, your goal is to find what aspects the
users liked about the books.
Look at the following example(s):
## EXAMPLE 1 START
Here are the user’s reviews of the books the user read:
[EXAMPLE REVIEW TEXTS]
Generate an abstractive user profile written in the
first-person narrative, based on the content of the book
as described in the reviews.
[EXAMPLE PROFILE]
## EXAMPLE 1 END
Here are the user’s reviews of the books the user read:
[REVIEW TEXTS]
Generate an abstractive user profile written in the
first-person narrative, based on the content of the book
as described in the reviews.
Generate a concise user profile in max 128 tokens total,
start right away with the user profile without introductory
text.
```

Trained Extractor Models

We train a model to extract user profiles from review texts, optimizing it for the recommendation objective—unlike the approaches discussed earlier, which leverage models trained for different goals. However, the challenge here is the absence of

ground truth user profiles, making direct supervision infeasible. Furthermore, the extractor model relies on an *argmax* operation for word selection, which prevents the flow of gradients during training. To address these challenges, we employ reinforcement learning (RL), using recommendation performance as the reward signal to guide the learning process.

RL-optimized Token Classifier. Given a single review as input, the model selects k words to maximize a ranking-based reward (e.g., DCG). The reward is computed over a pool of items, consisting of one positive item (the item corresponding to the review) and N randomly sampled negative items.

In this setup, the *state* is defined as the current profile, *actions* correspond to selecting tokens from the input review text, and the token classifier serves as the *policy network*, optimized using the REINFORCE Policy Gradient Method [Sutton et al., 1999].

At each step t , a word (action) a_t is sampled based on classifier probabilities and added to the user profile (state). The profile s_t is then passed through a frozen recommender model to compute the reward r_t . The episode terminates upon reaching a token budget B . To convert the token classifier into a word extractor, only the first token of each word is considered a valid action. Once a token is selected, all subsequent tokens of that word are automatically included, consuming from the budget.

The objective is to maximize:

$$J(\theta) = \sum_{t=1}^B r_t * \log p_{\theta}(a_t | s_t)$$

where $p_{\theta}(a_t | s_t)$ is the probability of selecting token a_t given the current profile s_t , as predicted by the token classifier.

To compute the reward signal, we consider two ranking metrics: Discounted Cumulative Gain (DCG) and Reciprocal Rank (RR). These metrics are defined as follows:

$$Reward_{DCG}(a_t) = \frac{1}{\log(rank + 1)} - \frac{1}{N + 2} - \text{best past reward}$$

$$Reward_{RR}(a_t) = \frac{1}{rank} - \frac{1}{N + 1} - \text{best past reward}$$

where *rank* denotes the position of the positive item in the ranking produced by the recommender model, given the current profile including the newly selected token. The worst possible reward (when the positive item is ranked last) is subtracted, with $N + 1$ representing the size of the ranked list. The best previous reward, is also deducted to encourage progressive improvement.

After training, the user profiles are constructed by selecting the highest-scoring words from all of the user’s training-time reviews. To finalize the profiles, we perform deduplication and truncate the profile to 128 tokens, consistent with the other profiling methods described earlier.

Anecdotal examples of the constructed user profiles, using all described methods, are provided in Tables 5.9 and 5.10.

5.2.2 Ranking Model

For item ranking, we use a two-tower architecture for representation learning—one tower for users and the other for items—following the common approach in neural information retrieval, where queries and documents/passages are encoded separately. The two towers are jointly trained with a shared loss function. The advantage of this bi-encoding approach lies in its efficiency, allowing for precomputation of encodings and the use of approximate vector search during inference. A pictorial overview is shown in Figure 5.2.

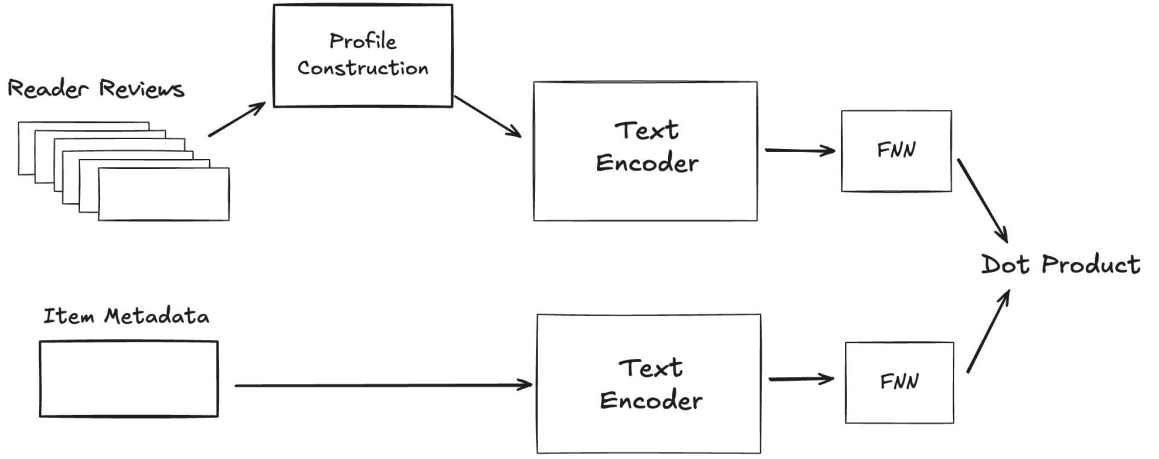


Figure 5.2: CUP architecture.

User profiles and item metadata are processed through a text encoder (e.g., BERT), followed by feedforward networks to learn latent representations. The resulting vectors are then compared via dot product to compute relevance scores, indicating the likelihood of a user liking an item. The *per-item* text typically includes book titles, category or genre labels, and a short description of the book’s content. The *per-user* text comprises the constructed profile derived from review texts (see Subsection 5.2.1).

Training

The user profile u consists of a sequence of text tokens $w_1^u \dots w_b^u$, where b denotes the token budget that constrains the input length (set to 128 in this work). The sequence is processed by the user tower, which comprises a text encoder followed by a feedforward neural network (FNN). The final user representation vector t^u is obtained by averaging the per-token embeddings.

The FNN consists of two layers with ReLU activation, computing the final user representation as

$$t^u = \text{ReLU}(t^u W_1^u + c_1^u) W_2^u + c_2^u$$

with an analogous formulation for items.

The score for a *user-item* pair is computed as

$$\hat{y}_{ui} = \sigma(\langle t^u, t^i \rangle)$$

where \langle, \rangle denotes the dot product and σ is the sigmoid function that converts scores into probabilities.

Training is performed using the Adam optimizer, minimizing the binary cross-entropy loss between predicted scores and ground-truth labels. Negative samples are drawn using a predefined sampling strategy (see Subsection 5.2.3).

$$\text{loss} = -(y_{ui} \log \hat{y}_{ui} + (1 - y_{ui}) \log(1 - \hat{y}_{ui}))$$

where y_{ui} is the gold label for the user-item pair.

Inference

Prediction for Ranking. At inference time, predictions are made for user-item pairs. The bi-encoder architecture enables pre-computation of representations for existing items and users. The scores for different test items, obtained via the final dot product, determine the *ranking* of candidate items. This approach ensures computational efficiency, following established practices in neural information retrieval (IR) [Lin et al., 2021].

Search-based Recommendation. In a deployed system, as opposed to controlled lab experiments, a typical usage mode is search-based re-ranking. A user provides context through a keyphrase query or an example of a previously liked item, functioning as a query-by-example. The expectation is to receive a ranked list of recommended items aligned with their *search intent*, rather than generic recommendations across all categories. An example user profile, along with both generic and search-based recommendations, is shown in Figure 5.3.

To achieve this, the system first retrieves approximate matches to the query (i.e., similarity-search neighbors) and then re-ranks a shortlist of, for example, the top-100 candidates. The CUP framework supports this mode by incorporating a lightweight BM25 retrieval model.

To evaluate the model in this setting, where ground truth query-user-item triples are unavailable, we search through all unlabeled items that match the category and textual description of the positive test instance. From this, we select the top 100 highest-scoring matches as the pool of candidates.

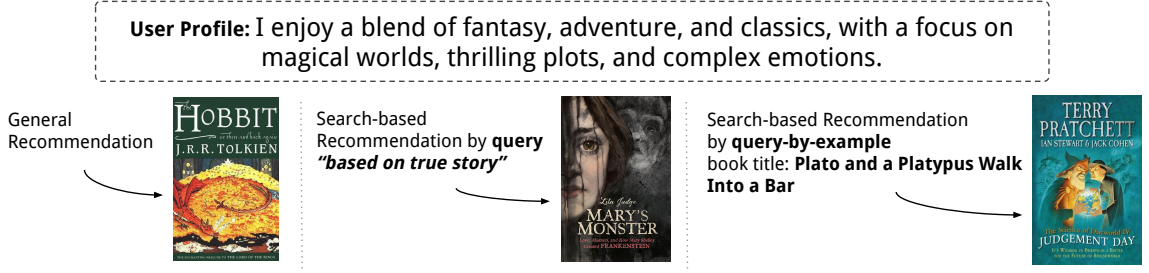


Figure 5.3: Example of search-based recommendation with text query and query-by-example.

5.2.3 Negative Training Samples from Unlabeled Data

Training in data-poor regimes presents a challenge due to the extreme imbalance between positive samples and unlabeled data points, particularly for sparse users. In many recommender applications, explicitly labeled negative samples—such as books rated poorly—are rare or nonexistent. This holds for the datasets used in this work. To address this, we introduce and experiment with various strategies for constructing negative training samples from unlabeled data:

Uniform Random Samples. Under the closed-world assumption (CWA), also known as Selected Completely At Random, negative training samples are drawn uniformly from all unlabeled data.

$$Q(j \mid u) = \begin{cases} 0, & \text{if } j \in I_u \\ \frac{1}{N - |I_u|}, & \text{otherwise} \end{cases}$$

where $Q(j \mid u)$ denote the probability that item j is selected as a negative sample for user u , I_u represents the set of items user u interacted with, and N is the total number of items.

This is a widely used standard approach.

Weighted Pos-Neg Samples. Works on PU (Positive & Unlabeled) learning for classification [Bekker and Davis, 2020], where only positive and unlabeled data are available, often treat unlabeled points as paired positive and negative samples with fractional weights. While these methods rely on learned estimates of class priors, their effectiveness diminishes in our setting due to the extreme class imbalance, i.e., the skew between labeled and unlabeled data.

In the recommendation domain, the closest approaches to PU learning are WMF ([Hu et al., 2008]) and EALS ([He et al., 2016]), which use a weighted loss function with distinct weights for positive and negative (unlabeled) instances. WMF assigns static weights to all positive and negative instances, while EALS refines the weight based

on item popularity. However, these approaches do not consider weighting schemes based on the specific user.

Our approach introduces a user-aware weighting mechanism by considering the relatedness of the potential negative sampled item j with the user’s positive training items I_u .

$$\text{relatedness}(j, u) = \frac{1}{|I_u|} \sum_{i \in I_u} \text{relatedness}(j, i)$$

To calculate the *item-item relatedness* ($\text{relatedness}(j, i)$), we leverage binary interaction data. Specifically, we use matrix factorization on the user-item interaction matrix (of the entire dataset) to compute a *relatedness score* for item pairs as the scalar product of their latent vectors, normalized by min-max scaling.

$$\text{relatedness}(j, i) = \frac{\mathbf{v}_j \cdot \mathbf{v}_i - \min}{\max - \min}$$

where v_j and v_i denote the latent vectors of items j and i computed by matrix factorization. The minimum (min) and maximum (max) values are estimated based on manual inspection of similarity scores between similar and dissimilar items.

Finally, each uniformly drawn unlabeled sample j is cloned into two instances, with one instance assigned a positive weight proportional to its average relatedness to the user’s explicitly positive items, while the negative clone’s weight is set to the complement.

$$Q(j^+ | u) = \text{relatedness}(j, u)$$

$$Q(j^- | u) = 1 - Q(j^+ | u)$$

This formulation ensures that unlabeled sampled items more related to the user’s history are more likely to be treated as positive examples, while those unrelated are more likely to be assigned negative labels, introducing a soft, user-aware weighting mechanism. In our experiments, we primarily use this method to train the recommender model.

Genre-based Samples. Category or genre tags of positive items can also guide the selection of negative samples, ensuring that negatives share at least one category with the corresponding positive item. Specifically, negative samples are drawn in proportion to the category distribution of the positive item.

Let C_j denote the set of categories associated with unlabeled item j , and C_i denote the set of categories associated with a positive item i from the user-item interaction

(u, i) . The probability of selecting a negative sample j is then proportional to the intersection of the categories of item j and item i .

$$Q(j \mid u, i) = \begin{cases} 0, & \text{if } j \in I_u \\ \frac{|C_j \cap C_i|}{|C_i|}, & \text{otherwise} \end{cases}$$

This method produces harder negative samples, as they are semantically closer to positive items. Moreover, it serves as an implicit training strategy for search-based scenarios, assuming categories function as proxies for queries in the absence of explicit query information.

5.3 Experiments

5.3.1 Rationale

As a difficult and less explored application area for recommenders, we investigate the case of book recommendations in online communities. These come with a long-tailed distribution of user activities, highly diverse user interests, and demanding textual cues from reviews and book descriptions.

Unlike in many prior works’ experiments, often on movies, restaurants or mainstream products, the data in our experiments is much sparser regarding user-item interactions. We design the evaluation as a *stress-test* experiment, with focus on text-rich users. With the focus on lightweight computation, we limited the input context to 128 tokens, hence the need for smart user profile extraction. Our experiments supports the choice of budget and architecture.

We further enforce the *difficulty of predictions* when items belong to groups with high relatedness within a group, by constraining disjointedness of authors per user in training and evaluation set. Thus, we rule out the near-trivial case of predicting that a user likes a certain book given that another book by the same author has been used for training.

5.3.2 Datasets

We use two book datasets from the UCSD recommender systems repository [McAuley, 2022], filtered for english reviews:

- **GR** [Wan and McAuley, 2018]: a Goodreads sample with item-user interactions for 1.5M books and 280K users, including *titles*, *genre tags*, *item descriptions*, *ratings* and *reviews*.
- **AM** [Ni et al., 2019]: an Amazon crawl for the books domain with 2.3M books and 3.1M users, including *titles*, *category tags*, *item descriptions*, *ratings* and *reviews*.

Figure 5.4 shows examples of book metadata from both datasets. As depicted, Amazon categories are more fine-grained than Goodreads genres, and book titles on Amazon often include additional details, making them significantly longer.

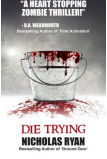

Goodreads Dataset	
	<p>Genres: Fiction, Fantasy Paranormal, Mystery Thriller Crime</p> <p>Author: Nicholas Ryan</p> <p>Title: Die Trying: A Zombie Apocalypse</p> <p>Description: Trapped amidst the horror of the zombie apocalypse, three men hear the sound of a helicopter overhead and know it represents their last desperate hope to survive the terror and reach safety. But survival comes at a cost - and the undead hordes that ravage the world are not the only lurking evil the men must face. (continued ...)</p>
Amazon Dataset	
	<p>Categories: Religion & Spirituality, Occult & Paranormal</p> <p>Author: Max Mason Hunter</p> <p>Title: Scary Ghost Stories: REAL Eyewitness Accounts: The Worlds Most Possessed Woods, Houses And Haunted Places (True Ghost Stories And Hauntings, True Horror Stories, Bizarre True Stories)</p> <p>Description: What would you do if you ever encountered a spirit or otherworldly entity? Do you think you would flee the scene, screaming as you put as much distance between you and the unknown? Or would you try your hardest to communicate with the entity? The idea of ghosts and spirits is not a new concept. (continued ...)</p>

Figure 5.4: Examples of item metadata from Goodreads and Amazon datasets.

Prior works mostly consider C-core data variants where all users and items have at least C interactions ($C=10$ or 5). This pre-processing focuses on interaction-based predictions, whereas our intention is to study the case of data-poor users and items. Instead, our data pre-processing is designed to evaluate text-based recommender system performance with text-rich users.

We view all book-user interactions with a rating of 4 or higher as positive, and disregard the lower ratings as they are rare anyway. We split the data into training, validation, and test sets (60:20:20), filtering out users with less than 3 items to guarantee at least one interaction per user in each set. We enforce disjointness of authors between the training and evaluation sets for the same user.

For the primary experiments in this chapter, we select 1,000 users from each dataset who, on average, write long reviews, based on the descending order of their average review length. These subsets are denoted as GR-1K-rich and AM-1K-rich, corresponding to each dataset, respectively. Other dataset slices are used in different subsections of the results section, where they are described in detail. Table 5.1 provides statistics for the datasets. Both 1K-rich variants are extremely sparse in terms of users sharing the same items, emphasizing the need to leverage text.

	#books	avg \pm stdv #u per book	#users	avg \pm stdv #b per user	avg \pm stdv review len
GR	1,573,290	6.54 \pm 55.95	279,969	36.77 \pm 93.25	178 \pm 259
GR-10K-dense	385,660	3.74 \pm 18.96	10,000	144.16 \pm 232.49	187 \pm 259
GR-10K-sparse	158,554	1.66 \pm 3.09	10,000	26.25 \pm 56.46	173 \pm 248
GR-1K-rich	45,412	1.17 \pm 0.65	1,000	53.32 \pm 91.04	426 \pm 384
AM	2,281,866	12.75 \pm 87.25	3,105,696	9.37 \pm 26.55	121 \pm 183
AM-10K-dense	203,930	2.51 \pm 7.3	10,000	51.19 \pm 177.37	239 \pm 281
AM-10K-sparse	58,443	1.36 \pm 1.6	10,000	7.93 \pm 11.19	106 \pm 169
AM-1K-rich	15,753	1.07 \pm 0.36	1,000	16.79 \pm 23.53	282 \pm 265

Table 5.1: Dataset statistics.

5.3.3 Experimental Setup

Metrics. Following the literature, we report NDCG@5 (Normalized Discounted Cumulative Gain) with binary 0-or-1 gain, P@1 (precision at rank 1), and ROC-AUC (Receiver Operating Characteristic-Area Under Curve). We compute these by micro-averaging over all test items of all users. Macro-averaged results over users followed similar trends. ROC-AUC is included as it evaluates the overall ranking quality; NDCG@5 reflects the observations that users care only about a short list of top-N recommendations; P@1 is suitable for recommendations on mobile devices (with limited UI). We also measured other metrics, like NDCG@k for higher k, MRR. None of these provides any additional insight, so they are not reported here.

Evaluation Modes. At test time, for each positive test item we sample 100 negative items from all unlabeled data. The system scores and ranks these 101 data points, creating a ranked list of items to be evaluated by the performance metrics introduced above. We evaluate all methods in two different modes with respect to the negative sampling strategy:

- **Standard:** sampling the 100 negative test points uniformly at random.
- **Search-based:** given the positive test item, searching for the top-100 approximate matches to the item’s description, using the BM25 scoring model.

Baselines. We compare our approach to several state-of-the-art baselines, which cover different methods, ranging from traditional collaborative filtering approaches to text-centric neural models and LLM-based rankers:

- **CF:** collaborative filtering operating on user-item interaction matrix by computing per-user and per-item vectors (dim=200) via matrix factorization [Funk, 2006].
- **LLMRank:** following [Hou et al., 2024], we use ChatGPT to rank the test items, given the user’s reading history. The history is given by the sequence of titles

of the 50 most recent books of the user, prefixed by the prompt “I’ve read the following books in the past in order:”. This prompt is completed by a list of titles of test-time candidate items, asking the LLM to rank them.

- **DeepCoNN** [Zheng et al., 2017] is a salient representative of using convolutional neural networks (CNN) over text inputs.
- **P5-profile**: prompting the T5 language model [Raffel et al., 2020], to provide a recommended item for a user, given their ids. Following [Geng et al., 2022], we train P5 using the prompts for *direct recommendation* to generate a “yes” or “no” answer. Pilot experiments show that the original method does not work well on sparse data. Therefore, we extend P5 to leverage review texts and item descriptions. Instead of ids, the prompts include item descriptions and sentences from reviews with the highest IDF scores (i.e., one of our own techniques).
- **BENEFICT** [Pugoy and Kao, 2020]: uses a frozen BERT model to create representations for each user review, which are averaged and concatenated to the item vectors. Predictions are made by a feedforward network on top. Following the original paper, each review is truncated to its first 256 tokens.
- **BENEFICT-profile**: our own variant of BENEFICT where the averaging over all user reviews is replaced by our IDF-based selection of most informative sentences, with the total length limited to 128 tokens (for comparability to CUP).

CUP Configurations. In the experiments, all CUP variants use an input budget of 128 tokens as a stress test, emphasizing our goal of limiting the computational and energy costs. The following CUP configurations cover different ways of user profile creation (see Subsection 5.2.1):

All CUP configurations below leverage full item metadata including title, genres, and description as item text, unless stated otherwise.

Extractive Methods:

- **CUP_{1gram}**: unigrams selected by TF-IDF scores.
- **CUP_{3gram}**: 3-grams selected by TF-IDF scores.
- **CUP_{idf}**: review sentences selected by IDF scores.
- **CUP_{sbert}**: review sentences selected by similarity to the corresponding item description, using Sentence-BERT.

Generative Models:

- **CUP_{kwT5}**: set of keywords generated by a fine-tuned T5 model¹.
- **CUP_{kwGPT}**: a keyword profile generated by ChatGPT (gpt-3.5-turbo).
- **CUP_{kwLlama}**: a keyword profile generated by Llama (Meta-Llama-3.1-8B-Instruct), given hand-crafted few-shot examples.
- **CUP_{absLlama}**: an abstractive 1st-person narrative profile generated by Llama, given hand-crafted few-shot examples.

¹<https://huggingface.co/ml6team/keyphrase-generation-t5-small-inspec>

Trained Extractor Models:

- **CUP_{dcgReward}**: words selected by token classifier scores, trained with discounted cumulative gain (dcg) reward.
- **CUP_{rrReward}**: words selected by token classifier scores, trained with reciprocal rank (rr) reward.

CUP Variants Using Item Metadata as User Profiles For comparison, we also configure a few CUP variants, utilizing item metadata (genre/category and title) to construct the user profiles.

- **CUP_{title}**: takes titles of all user’s training items as the user profile.
- **CUP_{tags}**: uses all genre tags from the user’s training items as the user text.
- **CUP_{basic}**: to observe the effect of item-side text, we restrict **CUP_{tags}** variant to use only the item title and genre as item text (removing the item description).

We used the following hyperparameters for the CUP configuration, obtained through grid search based on validation set performance: a batch size of 256, an FNN size of 200, and a learning rate varying between 4e-4 and 4e-5 depending on the specific configuration. During training of CUP models, we updated the FNN weights as well as the last layer of BERT, unless mentioned otherwise. All methods were run on NVIDIA Quadro RTX 8000 GPU with 48 GB memory, and we implemented the models with PyTorch.

5.4 Results

5.4.1 Comparison of CUP against Baselines

Standard Evaluation Mode. Table 5.2 shows the results for the AM-1K-rich and GR-1K-rich data, comparing our default configuration CUP_{idf} against all baselines, for the three different ways of sampling negative training points: Uniform, Weighted, and Genre-based (see 5.2.3). Results with statistical significance over the BENEFICT_{prof} baseline, by a paired t-test with p-value < 0.05, are marked with an asterisk (*). Bonferroni correction for multi-hypotheses testing is applied, reducing the test level of each pairwise comparison to 0.005.

We make the following key observations:

- The interaction-centric CF fails for this extremely sparse data. LLMRank also performs poorly. Solely relying on the LLM’s latent knowledge about books is not sufficient when coping with long-tail items. Popularity and position bias [Hou et al., 2024] further aggravate this adverse effect. To mitigate position bias, we ran a variant with smaller test sets of only 20 candidate negative items (per positive test item), as in the original setup of [Hou et al., 2024]. This boosts the NDCG@5 for LLMRank from 4.86% to 23.5%, on GR-1k-rich in standard evaluation, which

Method	Train Uniform			Train Weighted			Train Genre		
	NDCG@5	P@1	AUC	NDCG@5	P@1	AUC	NDCG@5	P@1	AUC
AM-1K-rich									
CF	3.06	1.0	49.05	2.88	0.69	49.93	2.94	1.0	47.74
LLMRank	4.62	2.27	51.99	n/a	n/a	n/a	n/a	n/a	n/a
DeepCoNN	3.0	0.8	50.2	3.01	0.83	50.2	3.0	0.83	50.2
BENEFICT	9.4	3.53	72.95	14.4	5.74	76.86	3.06	1.0	49.99
BENEFICT _{prof}	24.38	12.98	81.43	24.66	12.81	80.41	15.32	7.41	75.34
P5 _{prof}	24.9	14.5	81.64	n/a	n/a	n/a	n/a	n/a	n/a
CUP _{basic}	25.42	13.64	81.64	26.95*	14.27	79.9	13.07	5.63	75.04
CUP _{tags}	27.31*	14.99*	83.53*	28.83*	16.05*	82.16*	14.53	6.89	77.21*
CUP _{title}	29.07*	16.94*	<u>83.67*</u>	32.19*	18.6*	83.12*	18.07*	8.5	78.61*
CUP _{idf}	29.21*	15.82*	84.74*	<u>31.09*</u>	<u>17.71*</u>	83.23*	15.41	7.44	76.04
GR-1K-rich									
CF	4.44	2.69	30.23	3.83	2.26	48.18	2.78	1.54	30.97
LLMRank	4.86	2.1	51.5	n/a	n/a	n/a	n/a	n/a	n/a
DeepCoNN	10.45	4.02	75.13	5.4	1.86	57.65	4.52	1.44	53.21
BENEFICT	23.76	12.17	83.78	25.23	13.26	84.85	22.5	11.6	82.89
BENEFICT _{prof}	30.26	16.83	86.34	31.77	17.74	86.9	29.24	16.19	85.36
P5 _{prof}	28.01	14.46	85.86	n/a	n/a	n/a	n/a	n/a	n/a
CUP _{basic}	26.75	13.28	86.19	28.4	14.89	84.59	24.88	12.59	84.95
CUP _{tags}	30.92	16.3	88.35*	33.28*	17.83	86.77	29.75	16.28	86.97*
CUP _{title}	36.81*	20.8*	<u>90.15*</u>	39.82*	22.68*	90.29*	33.53*	19.27*	88.42*
CUP _{idf}	38.39*	<u>22.26*</u>	90.51*	<u>39.41*</u>	22.01*	90.17*	33.8*	18.87*	88.74*

Table 5.2: Standard evaluation. The best results are shown in **bold**, while the second-best results are underlined.

is still a large margin below CUP_{idf} reaching 39.41% (and similarly big gaps for the other dataset).

- The pre-transformers state-of-the-art text-based baseline DeepCoNN also performs very poorly. Post-transformers baseline BENEFICT is slightly better. Both of these baselines use the entire user review texts. At the same time, BENEFICT_{prof} and P5_{prof}, extended with our text-derived profiling, achieve decent performance.
- Between the four CUP configurations, we see a clear trend: review-based user profiling (*idf*) and title-based profiles (*title*) perform comparably, with each outperforming the other on different metrics and training strategies. Both consistently surpass user profiles based on genres or categories (*tags*), while simplifying item-side text by restricting it to titles and tags by removing item description (*basic*) yields the weakest performance. Remarkably, even CUP_{basic} outperforms most baseline models.
- The absolute numbers on GR-1K-rich are generally higher, due to the different data characteristics. The gains by CUP_{idf} over the baselines and over the tag-based CUP configurations are even more pronounced (e.g., outperforming BENEFICT_{prof} by 8 percentage points).
- Weighted negative sampling strategy for training almost always improves perfor-

mance although with small margin. However, genre-based negative sampling leads to a decline in performance across methods and metrics.

Method	Train Uniform			Train Weighted			Train Genre		
	NDCG@5	P@1	AUC	NDCG@5	P@1	AUC	NDCG@5	P@1	AUC
AM-1K-rich									
CF	3.02	1.0	49.16	3.03	0.83	49.76	3.02	1.03	48.34
LLMRank	3.49	1.1	50.75	n/a	n/a	n/a	n/a	n/a	n/a
DeepCoNN	0.05	0.0	17.82	0.05	0.0	17.83	0.05	0.0	17.81
BENEFICT	2.45	0.66	58.47	3.69	1.29	63.5	1.68	0.4	49.6
BENEFICT _{prof}	6.49	2.33	65.39	8.11	3.53	66.93	7.14	3.01	62.94
P5 _{prof}	8.4*	<u>3.88*</u>	67.89*	n/a	n/a	n/a	n/a	n/a	n/a
CUP _{basic}	7.13	2.76	68.4*	6.87	2.61	66.75	5.64	2.53	61.45
CUP _{tags}	7.41	2.93	70.17*	7.0	2.84	68.94*	4.68	2.01	62.73
CUP _{title}	9.84*	4.25*	<u>70.8*</u>	9.54*	<u>3.67</u>	70.75*	8.43*	3.76	67.26*
CUP _{idf}	8.93*	3.53*	71.84*	<u>9.14</u>	4.02	70.39*	5.98	2.3	62.83
GR-1K-rich									
CF	3.73	2.02	30.21	3.25	1.74	46.77	2.89	1.53	32.61
LLMRank	4.57	1.79	52.01	n/a	n/a	n/a	n/a	n/a	n/a
DeepCoNN	1.84	0.55	55.71	1.35	0.35	45.58	2.28	0.56	49.69
BENEFICT	6.73	2.38	69.7	6.88	2.44	71.06	6.74	2.21	69.02
BENEFICT _{prof}	8.95	3.4	72.01	9.63	3.59	73.23	9.56	3.81	72.32
P5 _{prof}	9.15	3.01	73.64*	n/a	n/a	n/a	n/a	n/a	n/a
CUP _{basic}	9.35	3.46	74.29*	9.84	3.55	72.9	9.38	3.36	73.41*
CUP _{tags}	10.66*	4.3*	76.46*	11.18*	3.94	75.6*	10.87*	4.08	75.4*
CUP _{title}	13.55*	<u>5.46*</u>	78.76*	13.02*	4.83*	<u>79.02*</u>	12.92*	4.98*	77.16*
CUP _{idf}	14.18*	5.9*	79.2*	<u>13.76*</u>	5.32*	78.94*	12.7*	5.04*	77.18*

Table 5.3: Search-based evaluation. The best results are shown in **bold**, while the second-best results are underlined.

Search-based Evaluation Mode. Table 5.3 presents results for the more challenging search-based evaluation mode. As expected, absolute performance is significantly lower, highlighting the difficulty of this realistic setting.

Among all methods, CF has the least degradation compared to standard evaluation (despite still underperforming other models). This is due to its lack of reliance on text, unlike other approaches. Nevertheless, the relative comparisons between models are nearly identical to those in the standard evaluation. In this evaluation mode, CUP_{idf} generally outperforms CUP_{title}, except for one metric in the AM dataset.

The performance gap between genre-based negative sampling and other strategies narrows, as genre-based sampling aligns more closely with search-based evaluation. Finally, the AM dataset benefits more from weighted training than the GR dataset in this mode.

Method	ALL	u-s	u-r	u-b	s-s	s-r	s-b
AM-1K-rich							
CUP _{idf}	<u>9.14</u>	5.93	7.09	<u>12.0</u>	8.6	11.71	12.78
CUP _{sbert}	9.0	5.19	7.32	11.46	9.44	14.53	14.23
CUP _{1gram}	9.08	6.24	7.14	11.21	9.76	17.0	13.25
CUP _{3gram}	8.98	5.54	7.01	11.81	8.2	13.97	10.99
CUP _{kwT5}	9.5	6.03	7.15	12.48	8.71	12.38	17.78
CUP _{kwGPT}	8.93	5.95	6.95	11.59	8.29	11.3	13.85
CUP _{kwLlama}	9.01	<u>6.23</u>	6.9	11.82	7.83	11.15	12.45
CUP _{absLlama}	8.04	5.14	5.86	10.63	5.23	<u>16.11</u>	11.94
CUP _{dcgReward}	8.69	6.18	6.87	11.1	<u>10.13</u>	10.8	9.55
CUP _{rrReward}	8.95	5.41	<u>7.24</u>	11.1	10.61	15.91	<u>14.85</u>
GR-1K-rich							
CUP _{idf}	13.76	7.32	11.42	14.96	17.37	17.43	19.56
CUP _{sbert}	13.51	8.47	10.41	15.05	16.14	16.85	19.79
CUP _{1gram}	13.47	7.82	10.61	14.64	16.14	17.95	20.32
CUP _{3gram}	13.19	7.3	10.47	<u>15.08</u>	14.06	16.16	17.94
CUP _{kwT5}	<u>13.76</u>	7.51	<u>11.03</u>	14.43	17.43	19.33	<u>22.17</u>
CUP _{kwGPT}	13.78	7.97	10.56	14.33	<u>18.32</u>	20.36	23.18
CUP _{kwLlama}	13.54	<u>8.06</u>	10.63	15.25	13.88	16.83	19.54
CUP _{absLlama}	13.22	6.75	9.87	14.2	18.76	<u>19.59</u>	21.37
CUP _{dcgReward}	13.16	7.44	10.27	14.4	15.23	17.64	20.12
CUP _{rrReward}	13.15	7.12	10.33	14.57	14.06	16.73	20.49

Table 5.4: CUP results, by user/item groups (NDCG@5 with Search-based evaluation). The best results are in **bold**, while the second-best results are underlined.

5.4.2 Comparison of CUP Configurations

For insight on specific groups of users and items, we split the 1000 users and their items into the following groups, reporting NDCG@5 for each group separately. Note that this refinement drills down on the test outputs; the training is unaffected.

- Items are split into **unseen (u)** and **seen (s)** items. Unseen test items have not been seen at training time. Seen test items appeared as positive training items for a different user.
- Users are split into groups based on the number of books per user distribution:
 - **Sporadic (s)** users are the lowest 50% with the least numbers of books. For GR-1K-rich, this threshold is 13 books per user; for AM-1K-rich it is 5 (with means 6 and 3, resp.).

- **Regular (r)** users are those between the 50th percentile and 90th percentile, which is between 13 and 71 books per user for GR-1K-rich, and between 5 and 20 for AM-1K-rich (with means 31 and 9, resp.).
- **Bibliophilic (b)** users are the highest 10%: above 75 books per user for GR-1K-rich and above 20 for AM-1K-rich (with means 156 and 43, resp.).

Table 5.4 shows NDCG@5 (for all and per item/user group) with search-based mode, comparing all CUP configurations with weighted training. We offer the following notable observations:

- Across all groups, all CUP configurations are competitive. The overall differences between them are relatively small. The winner, by a small margin, is CUP_{kwT5} for AM and CUP_{kwGPT} for GR datasets, closely followed by the default configuration CUP_{idf} and $CUP_{kwLlama}$ as well as CUP_{sbert} . None of the methods is able to extract the “perfect” gist from the noisy review texts; but all of them do a decent job. Despite the fact that generated profiles are slightly ahead of the others, the bottom line is that a relatively simple configuration, like IDF-selected sentences, is a good choice.
- The CUP_{kwGPT} variant achieves its highest gains for the richer item/user groups: seen items and regular or bibliophilic users (GR dataset). This provides ChatGPT with longer and more informative texts. A similar effect, but to a lesser and noisier extent, can be observed for T5-based CUP_{kwT5} . Conversely, these methods perform substantially worse on the sporadic-unseen group.
- The learned profiles $CUP_{dcgReward}$ and $CUP_{rrReward}$ perform similarly on the unseen items for both datasets. While for the AM dataset seen items, we observe that the *rr* reward performed better than the *dcg*. Our anecdotal examples in Tables 5.9 and 5.10 show that these rewards extract different words from the reviews.

Table 5.5 presents results for comparing different CUP configuration for standard evaluation mode. We observe similar patterns to the search-based evaluation (Table 5.4), with CUP_{kwT5} performing the best overall, followed by $CUP_{kwLlama}$ and CUP_{sbert} .

Method	ALL	u-s	u-r	u-b	s-s	s-r	s-b
AM-1K-rich							
CUP _{idf}	31.09	23.58	27.75	36.84	30.58	41.04	27.77
CUP _{sbert}	31.18	23.96	29.7*	35.22	30.45	37.34	32.26
CUP _{1gram}	31.13	23.72	28.24	36.0	32.21	42.09	32.99
CUP _{3gram}	30.6	23.02	27.65	<u>36.05</u>	29.02	38.48	31.4
CUP _{kwT5}	31.55	25.98*	<u>29.04</u>	35.97	25.81	38.38	<u>34.22</u>
CUP _{kwGPT}	30.82	23.52	28.59	35.21	30.95	38.67	34.8
CUP _{kwLlama}	<u>31.26</u>	<u>25.43</u>	29.83*	35.12	28.91	37.66	25.35
CUP _{absLlama}	30.35	24.7	27.64	34.7	29.31	39.81	29.03
CUP _{dcgReward}	30.89	24.53	28.29	35.36	<u>32.2</u>	<u>41.63</u>	26.81
CUP _{rrReward}	31.03	24.55	28.33	35.5	31.56	40.85	31.44
GR-1K-rich							
CUP _{idf}	39.41	30.09	36.72	41.95	38.22	40.51	46.65
CUP _{sbert}	<u>39.82</u>	30.87	37.08	42.71*	37.09	39.56	46.82
CUP _{1gram}	39.24	29.74	36.44	41.34	<u>38.82</u>	40.85	48.82*
CUP _{3gram}	39.2	30.52	<u>37.2</u>	41.98	35.53	37.88	44.68
CUP _{kwT5}	40.1*	29.35	37.74*	42.16	38.54	<u>42.45</u>	<u>49.2*</u>
CUP _{kwGPT}	39.79	29.27	36.73	41.86	39.4	43.23*	50.13*
CUP _{kwLlama}	39.34	30.21	36.5	<u>42.19</u>	35.95	39.26	47.24
CUP _{absLlama}	38.44	28.88	35.13	40.72	37.57	41.4	48.17
CUP _{dcgReward}	39.53	31.02	36.99	41.89	37.35	40.27	46.93
CUP _{rrReward}	39.32	<u>30.96</u>	36.88	41.66	36.82	38.76	47.62

Table 5.5: CUP results, by user/item groups (NDCG@5 with Standard evaluation). The best results are in **bold**, while the second-best results are underlined.

5.4.3 Efficiency of CUP

Two architectural choices make CUP efficient:

- input length restricted to 128 tokens, and
- fine-tuning only the last layer of BERT and the FNN layers.

For more analysis, we measured the training time and resulting NDCG on the GR-1K-rich data, comparing different input size budgets and choices of tunable parameters. Figure 5.5 shows the NDCG@5 results on the validation set.

We observe that the 128-token configuration has the lowest training cost: significantly less time per epoch than the other variants and fast convergence (reaching its best NDCG already after 15 epochs in ca. 3000 seconds). The 256- and 512-token models eventually reach higher NDCG, but only by a small margin and after much longer training time.

As for the number of trainable parameters, we observe that the variant with frozen

BERT takes much longer to converge and is inferior to the preferred CUP method even after more than 50 epochs. The other extreme, allowing all BERT parameters to be altered, performs best after enough training epochs. However, it takes almost twice as much time per epoch. From the benefit/cost perspective, our design choice hits a sweet spot in the spectrum of model configurations.

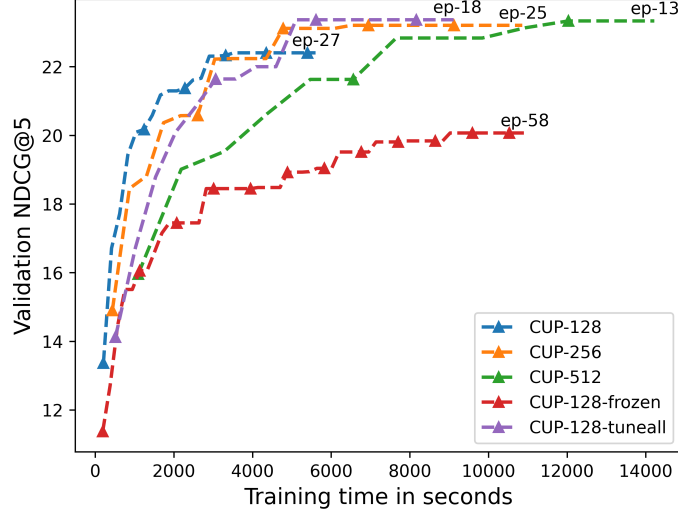


Figure 5.5: Training time for different input lengths and trainable parameters (lines are marked every 5th epoch).

5.4.4 Influence of Interaction Density

The slices AM-1K-rich and GR-1K-rich are constructed with a text-centric stress-test in mind. Both are extremely sparse in terms of user-item interactions. To study the influence of sparseness in isolation, we created two larger samples of both datasets, one still sparse by design and the other denser in terms of users sharing items. We refer to these as AM-10K-sparse and AM-10K-dense, and analogously for GR. All these samples cover 10K users: 10x more users than our previous text-rich slices, to allow more potential for learning from interactions. Table 5.1 shows the statistics.

The datasets are constructed as follows. To ensure connectivity in the interaction graphs, we first select 500 users and sample 2000 books connected to these users, both uniformly at random. This is the seed for constructing two variants of data, based on the cumulative item degrees of users, that is, the sum of the #users per book for all books that the user has in her collection:

- **10K-dense:** We randomly select 10K (minus the initial 500) users from all users in proportion to the users' cumulative item degrees. This favors users with many or popular books.
- **10K-sparse:** We select users inversely proportional to their cumulative item degrees, thus favoring sparse users.

In this sensitivity study, we focus on comparing CF (based on matrix factorization) against our default CUP configuration (CUP_{idf} with IDF-selected sentences) and the hybrid combination CUP + CF. In the hybrid setting, we enhance text-based representations with collaborative filtering (CF) signals, by concatenating the learned latent per-user and per-item vectors. These are precomputed by factorizing the training-set user-item interaction matrix, using the method of [Funk, 2006], with gradient descent for minimizing a cross-entropy loss.

Method	Standard Evaluation							Search-based Evaluation						
	ALL	u-s	u-r	u-b	s-s	s-r	s-b	ALL	u-s	u-r	u-b	s-s	s-r	s-b
AM-10K-Sparse														
CF	6.35	0.0	0.04	0.02	22.84	21.89	16.03	5.06	0.0	0.01	0.01	18.79	17.16	12.39
CUP	32.53	19.06	25.56	36.05	39.75	44.72	45.80	9.78	3.40	3.90	6.64	22.43	21.65	18.29
CUP+CF	13.26	11.66	6.28	1.47	35.29	31.55	20.77	5.79	1.65	0.43	0.13	19.45	18.77	13.00
AM-10K-Dense														
CF	33.32	0.08	0.04	0.20	68.76	56.25	43.05	19.02	0.13	0.03	0.14	51.76	34.75	21.46
CUP	47.26	11.55	19.44	27.87	68.95	62.59	53.96	20.24	1.23	2.44	5.81	49.98	34.40	21.12
CUP+CF	33.66	2.20	0.30	0.52	69.00	56.64	43.34	19.12	0.38	0.04	0.27	51.70	35.04	21.49

Table 5.6: NDCG@5 for AM-10K Sparse and Dense datasets under Standard and Search-based evaluations across user/item subgroups.

Method	Standard Evaluation							Search-based Evaluation						
	ALL	u-s	u-r	u-b	s-s	s-r	s-b	ALL	u-s	u-r	u-b	s-s	s-r	s-b
GR-10K-Sparse														
CF	20.02	0.0	0.0	0.02	55.96	46.36	37.29	15.71	0.0	0.0	0.01	46.44	36.62	28.0
CUP	44.62	24.48	32.59	38.31	56.32	58.01	53.98	17.97	3.9	5.79	8.89	37.29	32.64	26.18
CUP+CF	26.92	18.58	11.25	2.63	60.98	57.24	41.46	16.4	2.25	0.65	0.07	42.91	39.37	29.3
GR-10K-Dense														
CF	47.36	0.01	0.01	0.0	73.9	64.76	50.4	39.73	0.02	0.0	0.01	64.81	55.22	40.38
CUP	52.47	22.14	23.22	27.15	69.19	62.81	53.34	36.42	4.66	5.18	7.38	56.91	48.24	36.34
CUP+CF	49.59	5.47	1.9	0.59	74.35	66.66	54.1	41.22	0.29	0.04	0.04	65.48	56.76	43.03

Table 5.7: NDCG@5 for GR-10K Sparse and Dense datasets under Standard and Search-based evaluations across user/item subgroups.

Tables 5.6 and 5.7 show the results for the AM-10K data and GR-10K data, comparing the sparse vs. the dense variants with standard and search-based evaluation modes. Key insights are:

- As the 10K-sparse data is already much denser than the stress-test 1K-rich slices, CF can achieve decent results on the sparse data. Especially in search-based evaluation, CF is almost on par with CUP. Note that these gains come from the seen items alone, as CF is bound to fail on unseen items.

- CUP is still the clear winner on the sparse variant. In standard evaluation, it achieves 2 to 3 times higher NDCG@5 than CF. The hybrid configuration, with text and CF vectors combined, is inferior to learning from text alone.
- For the 10K-dense data, CF alone performs well, as this is the traditional regime for which CF has been invented. An interesting point arises for the search-based mode and with dense data. Here, the negative test points are closer to the positive sample (after the BM25 search), and pose higher difficulty for text cues alone to discriminate them. Thus, CF becomes more competitive. On GR-dense, hybrid CUP+CF performs best.

5.5 Analysis of User Profiles

Desiderata for an ideal profile are i) *high utility*: leading to strong recommender performance, ii) *easy interpretability*: supporting humans in understanding the gist of somebody’s interests, and iii) *sound faithfulness*: capturing the user’s style in writing reviews. Clearly, there are trade-offs between these dimensions. In terms of utility, our experiments show that several kinds of profiles are roughly on par, with some simple ones being slightly ahead. On the other hand, the most interpretable profiles are the generative ones using an LM. Finally, the extractive profiles, like salient sentences from reviews, appear most faithful. For illustration, Tables 5.9 and 5.10 provide examples of different kinds of profiles from Amazon and Goodreads datasets, respectively. Sample of original raw reviews from these users are shown in Tables 5.11 and 5.12.

To obtain more insights, we computed various statistics: distributions of part-of-speech (POS) tags (i.e., word categories) and distributions of IDF-weight mass among frequent words. These are derived by concatenating profiles of all users, for each of the most interesting profile types, including the users’ original reviews.

The statistics are shown in Table 5.8. The first column denotes the fractions of the three most frequent POS types. We observe that the LM-generated profiles have a much higher share of nouns, which are more informative than verbs or adjectives/adverbs. The second column shows the average IDF weight for the words in the 20-th percentile of the word frequency distribution. We observe that the original reviews carry low IDF weight, reflecting their verbose and noisy nature. In contrast, the generative profiles select high-IDF words as most informative cues. Finally, the third column shows the Kullback-Leibler divergence (with Laplace smoothing, $\alpha = 0.1$) for the user profiles generating the book descriptions (with removal of stop-words). We observe that the vocabulary of original reviews differs significantly from the wording in book descriptions, whereas IDF-aware profiles and keyword-generated profiles are much closer to the vocabulary that describes book contents.

	Amazon					Goodreads				
	POS			avg. IDF	KL-div	POS			avg. IDF	KL-div
	NN	VB	AD			NN	VB	AD		
all reviews	0.25	0.19	0.17	0.009	3.28	0.28	0.19	0.16	0.008	2.93
IDF sentences	0.33	0.16	0.18	0.008	1.72	0.39	0.15	0.19	0.011	1.19
Llama abstractive	0.35	0.15	0.18	0.065	1.7	0.36	0.14	0.19	0.084	1.31
Llama keywords	0.67	0.05	0.23	0.42	1.55	0.68	0.04	0.23	0.503	1.27
unigrams	0.47	0.23	0.25	0.537	1.99	0.54	0.18	0.22	0.921	1.22

Table 5.8: Statistical comparison of different kinds of user profiles

5.6 Related Work

Exploiting User Reviews. Incorporating user-provided reviews into recommenders has been pursued with deep neural networks [Chen et al., 2018, Liu et al., 2019, Wu et al., 2019, Wu et al., 2021b, Zhang et al., 2017, Zheng et al., 2017] and latent-factor models [Hu et al., 2019, Peña et al., 2020, Shalom et al., 2019]. Some works augment collaborative filtering (CF) models with user text, to mitigate data sparseness (e.g., [Liu et al., 2020a, Lu et al., 2018, Shalom et al., 2018]). [Wang et al., 2021] proposes to learn the importance of review meta-data (age, length etc.), but textual content is disregarded. Other works like [Shuai et al., 2022] incorporate the similarity of user reviews and item descriptions into graph-based learning. Mostly pre-dating the advent of large language models, these methods have been found to have only limited effects [Sachdeva and McAuley, 2020].

Recent work by [Ramos et al., 2024] takes advantage of large language models to generate short user profiles from reviews. Our framework subsumes this approach as a special case. The brand-new work of [Dong et al., 2025] includes reviews in learning a model over a shared latent space. In contrast, our framework makes user profiles explicit before feeding them into the recommender, giving the user a chance to inspect and edit this personal data.

Exploiting Language Models. Pre-trained LMs can be leveraged to i) encode item-user signals into transformer-based embeddings, ii) infer recommended items from rich representations of review texts, or iii) implicitly incorporate the latent “world knowledge” of the LM.

A representative of the first line is P5 [Geng et al., 2022], which employs prompt templates for the T5 language model. We include an enhanced variant of the P5 method in our experiments. The recent work of [Ramos et al., 2024] generates user profiles by prompting LLMs like Llama and Mistral. The profiles are used for rating prediction, not for ranking a larger pool of candidate items. The method is designed for short reviews; text-rich book reviews are not considered.

On the second direction, the works of [Pugoy and Kao, 2020, Pugoy and Kao, 2021] use BERT to create representations for user and item text, using short chunks of single reviews as the unit of granularity. The methods then aggregate these per-review vectors either by averaging [Pugoy and Kao, 2020] or via k-means clustering [Pugoy and Kao, 2021]. Our experiments include BENEFICT [Pugoy and Kao, 2020] as a baseline.

On the “world knowledge” direction, early works, using BERT, elicit knowledge about movie, music and book genres [Penha and Hauff, 2020]. Recent works prompt large language models (LLMs), such as GPT or Llama, to generate item rankings for user-specific recommendations [Hou et al., 2024, Wang and Lim, 2023] or predict user ratings [Kang et al., 2023], in a zero-shot or few-shot fashion. Our experiments include [Hou et al., 2024] as an LLM-powered baseline.

Supporting the Long Tail. Support for long-tail items and users falls under the theme of cold-start and zero-shot recommendations (e.g., [Cao et al., 2020, Li et al., 2019, Liu et al., 2022, Luo et al., 2023, Raziperchikolaei et al., 2021, Zang et al., 2023]). State-of-the-art methods are reasonably successful on new items, by embedding the item features into the same space as warm items, thus learning relatedness between warm and cold items. This assumes that cold items come with tags and descriptions. For the user side, this assumption is not practical: users would not likely expose a rich profile when they are new to a community or merely occasional contributors. In this data-poor regime, the only option is to harness textual cues from a small number of reviews.

Weighted Negatives and Negative Sampling. Previous works on weighted negatives have focused on using static weights for training recommender models from implicit data. For example, [Hu et al., 2008] tuned two hyperparameters per dataset—one for the positive instances and one for the negative instances. Additionally, [He et al., 2016] adjusted the weights for negative instances according to their popularity. The rationale behind this approach was that popular items are more likely to be shown to users, but despite this, they may not have been interacted with, and therefore should be given higher weight as negative instances.

State-of-the-art approaches to sampling negative training points in recommenders pursued an iterative paradigm where the samples are dynamically adapted so that points with higher model scores receive more weight [Zhang et al., 2013, Chen et al., 2022] (see [Chen et al., 2023] for more details on negative sampling and non-sampling strategies in recommender systems). These techniques incur higher computational costs and are agnostic to the structured aspects of the data, like categories and other metadata, which our work leverages.

5.7 Conclusion

In this chapter, we investigated review-based user profiling for personalized, search-based recommendation. We introduced CUP, a transformer-based framework that constructs concise user profiles by judiciously selecting informative pieces of user text. Beyond standard recommender system evaluation, we simulated a search-based interaction mode, reflecting a more realistic and dynamic system usage scenario. Additionally, we explored alternative strategies for sampling negative training instances. Our experiments demonstrate that leveraging user text is particularly beneficial in data-poor regime and that creating a concise profile clearly outperforms aggregating all available user text without distinguishing between relevant information and noise. Moreover, concise profiling offers the added benefit of lower energy and computational costs, making it a more efficient choice in practice. Among the various CUP configurations, we found that most performed similarly, suggesting two practical options: (1) selecting the lowest-cost variant, which utilizes IDF-based n-grams or sentence-level review excerpts, or (2) opting for LLM-generated profiles, which come at a higher computational cost but offer enhanced human readability.

Method	User Profile
genres	africa, nature ecology, americas, history, europe, leaders notable people, relationships, world, science math, historical, biographies memoirs, self - help, genre fiction, literature fiction
title	The Pirate Coast: Thomas Jefferson, The First Marines, And The Secret Mission Of 1805. The Story of the Irish Race. Horse Sweat and Powder Smoke: The First Texas Cavalry in the Civil War. God's Battalions
IDF sentences	Confederate Navy Raider. Irish history - in a nutsell!. Heroes, US Marine Corps Medal of Honor Winners by Marc Cerasini. Women's Options on the American Fronteir. Norwegian Immigration 1850s.
SBERT sentences	It was actually mostly written by Elisabeth Koren, wife of Reverend U. V. Koren. The area of research is the Arkansas Missouri Borderlands. Story is of a woman radical whose life was brought up short by Senator
unigrams	texas, navy, colt, marines, koren, norwegian, quege, ranger, villhelm, norwegians, nutsell, leonto, markist, rangers, caliber, fronteir, cerasini, korens, book, cavalry, laundress, pistols, xo, agnes, praire, outstanding
T5 keywords	norweger immigration 1850s book, texas, texas, funeral. amateur historian, history buffs, acoustic borderlands, ignoble deeds, south. socialist woman, gender issues, birth control, abortion, labor unions, health care
ChatGPT keywords	norwegian immigration, 1800s life, historical commentary, civil war, confederate navy, mexican war, texas cavalry, american frontier, lost states, irish history, texas rangers, mexican war, agnes smedley, us marine corps
Llama keywords	Norwegian Immigration, 1850s, historical reference, Civil War, eyewitness accounts, Arkansas Missouri Borderlands, Confederate Navy Raider, Mexican War, Fort Brown, Western Frontier, Simon Kenton
Llama 1st-person	I'm a history buff with a passion for non - fiction books, particularly those that delve into the lives of ordinary people during extraordinary times. I enjoy reading about the experiences of women, immigrants
learned dcg reward	texas, confederate, frontier, towns, economic, marines, norwegian, marine, ship, 1800s, irish, norwegians, across, code, xo, 1970, cavalry, political, expedition, war, springing, whose, teacher, minnesota, mexican
learned rr reward	texas, lust, v, minnesota, studying, war, woman, xo, towns, economic, change, praire, life, clinical, navy, mexican, state, mow, tv, scenery, crusades, willhelm, buy, laundress, wilkes, true, expedition, opportunities

Table 5.9: User profiles constructed by various methods from **Amazon** dataset (truncated to max 3 lines). The original user reviews are shown in Table 5.11.

Method	User Profile
genres	mystery thriller crime, fiction, romance, history historical fiction biography, fantasy paranormal, non - fiction, children, young - adult, comics graphic
title	When Somebody Kills You. The Night Before (Hubbard's Point/Black Hall Series). A Killer Read (Ashton Corners Book Club Mystery #1). Much Ado About Muffin (Merry Muffin Mystery, #4). Murder of
IDF sentences	Fans of Erika Chase or Kylie Logan will enjoy the latest in Laura DiSilverio's Readaholics series. Welcome back, Hercule Poirot! Readers who enjoy Elaine Viets will enjoy a trip to Fernglen Galleria. I received this
SBERT sentences	She has a rare condition called "Cotard's Syndrome" which is both a help and a danger to Fiona as she goes undercover to help solve her latest case. Boston Homicide Detective Jane Rizzoli and Medical Examiner
unigrams	netgalley, her, book, characters, pru, likeable, mystery, ezine, she, rinette, murder, fans, jenna, cozy, installment, jaymie, series, aroostine, jeffries, pargeter, jaine, honest, josie, mysteries, story, enjoy, detective
T5 keywords	Sehr geehrte Damen und Herren! cozy mystery authors, american, gardener, gardener, cottages, wrought iron irony, wrought. legal thrillers, national guard, family history, spiritual convictions, narrative. tv series
ChatGPT keywords	clara quinn, finn's harbor, maine, occult bookstore, murder investigation, psychic abilities, small - town setting, friendship, suspects, psychic gift, murder trap, maine setting, cozy mystery, tv production company
Llama keywords	mystery, cozy mystery, police procedural, detective fiction, amateur sleuth, whodunit, thriller, suspense, crime fiction, murder mystery, courtroom drama, legal thriller, romance, romantic suspensegothic
Llama 1st-person	I'm a fan of cozy mysteries with amateur sleuths, often women, who solve crimes in small towns or villages. I enjoy books with complex plots, unexpected twists, and surprising endings. I appreciate stories
learned dcg reward	emily, texas, lucy, china, dani, lizzie, thomas, kate, josie, georgia, jo, jeremy, aubrey, elsie, christmas, fiona, forensic, jenna, rinette, hank, austen, sheila, detective, manhattan, rachel, river, charlotte, bayles
learned rr reward	china, kate, texas, josie, chilling, charlotte, lucy, polished, emily, knight, aurora, caprice, fisk, rinette, rachel, rina, residents, forensic, bill, fiona, mel, dani, police, sheila, hannah, death, recipes, jenna, critical, sunday

Table 5.10: User profiles constructed by various methods from **Goodreads** dataset (truncated to max 3 lines). The original user reviews are shown in Table 5.12.

Book Title	User Review
The Diary of Elisabeth Koren	Norwegian Immigration 1850s. Book was in as good a condition as stated. This book came to me from Missouri...It is an outstanding commentary on life in about the 1850s, from leaving Norway by ship, crossing the Atlantic, to arriving in New York to going to Decorah, Iowa...my driving reason for this book was a taste of life of the 1800s, a possible connection between the Korens and my forbears...it is a good read and spells out the trials of life of their times...
Borderland Rebellion	History, as it should be told! As an amateur historian, I dislike some of the newer works we often see that "sanitize" history to that the writer thinks today's readers might enjoy and select "facts" seen through today's eye to justify one side or the other for their own purposes...efforts to show both sided of the Civil War via the eyewitnesses is OUTSTANDING!...While this was a back-water conflict when compared to the battles being fought in the East...
The Training Ground: Grant, Lee, Sherman, and Davis in the Mexican War	Excellent! This book covers the Mexican War from Fort Brown down to Mexico city and shows that a number of the young officers who would later be Generals in the American Civil War.
Last Flag Down: The Epic Journey of the Last Confederate Warship	This is a great book dealing with historical fact and was written from the perspective of the Ship's Executive Officer (XO)...My recommendation is that if your buy this one, you should also buy Sea of Gray as Sea of Gray covers...This is a rare glimpse of a force that never reached 1000 men total in the entire Civil War...Both books have their different perspectives and complement each other. Neither book is "complete" with out the other though they give that...
Norwegians on the Prairie: Ethnicity and the Development of the Country Town	Life on the Praire. Interesting read. This is about small towns springing up across Minnesota and how these towns contributed to the social, moral and economic makeup of what would become the state of Minnesota and the high cost of living isolated lives and other factors associated with prairie towns.
The Story of the Irish Race	Irish history - in a nutsell! The Story of the Irish Race. I first read this book from the library. Mine that I bought is in far better condition than the one I checked out. Outstanding read and if one wants to get the lowdown on this history of the Irish/Celts/Scots this is by far one of the more concise and understandable books on the history. Keep in mind, I am not prone to just handing out 5 stars, this is in outstanding condition, its an outstanding...
I Do: Courtship, Love Marriage on the American Frontier	Of interest to all studying life on the frontier. This is one of those books that one can spend a few minutes reading and then put aside to go to the store or mow the yard. This is a series of short but true stories...One "bride" was left in 1861 on the East Texas prairie by her husband in a sod house, with a quart of black-eyed peas to feed herself and two children for at least 3 days after he set out to get a job after having gambled away two barrels of flour and the rest of the families food supplies...There are many such stories...very worthy read.

Table 5.11: Example of user-written reviews from the **Amazon** dataset (for reviews exceeding 6 lines, sentences from different parts of the review are selected to fit within the table).

Book Title	User Reviews
The Vanishing	During a seance in 1865, the experienced medium Seraphina, senses something unusual going on during this session at Havenwood. Instead of the usual pleasant jumble of voices from the loved ones of those attending the seance, there is only silence followed by a rumbling that soon fills the room with darkness and evil...I loved this modern ghost story with its old-fashioned gothic feel...The book was very suspenseful, with a touch of old-fashioned romance.
Claws for Alarm	Not quite as good as the first book in the series, but still a great read for cat lovers who prefer intelligent, tough cats in their mysteries, as opposed to cutesy. I look forward to reading more Nick and Nora adventures.
Bookman Dead Style	Things get more exiting in the small town of Star City, Utah when the annual Star City Film Festival comes to town...Although happily dating her scientist boyfriend Seth, she has an instant, platonic connection to Matt Bane, one of the visiting movie stars. Soon, at the handsome actor's request, Clare finds herself in the middle of a murder investigation as she tries to prove that Matt is innocent of killing his sister...This book has some great moments, but...
Sleeping in the Ground	A wedding between a high profile bride and groom in Yorkshire has been given a lot of media coverage, and many people are interested in their upcoming nuptials. However, this fairytale wedding turns into a nightmare when the wedding party and their guests are caught in the path of gunfire...There are many types of books falling under the category of "Mystery" and I enjoy most of them. Of all the types of mysteries, there is something about a good police...
Mrs. Jeffries and the Merry Gentlemen	...If this is the first time you have read one of these books, the premise of the series may sound far-fetched. Scotland Yard Inspector Gerald Witherspoon solves murder cases with help of his housekeeper Mrs. Jeffries, other staff members, and close friends of the household. The catch is Witherspoon doesn't realize they are helping him!...The book has an interesting plot, but it's the characters that really make this series great...pleasure to spend time with...
That Last Weekend: A Novel of Suspense	I have enjoyed this author's cozy mysteries for quite some time and I have started reading her newer standalone thrillers, and I really like them! "That Last Weekend" is my favorite so far of all her books... There is a fairly large cast of characters to keep track of and I was sometimes confused about whether the events being described happened in the past or the present...this book is full of unpredictable plot turns that kept me guessing until the end...
In the Shadow of the Glacier	Interesting police procedural with an experienced veteran detective partnered with a rookie beat officer. I like their interactions although the sergeant is sometimes a little too harsh on the new officer, Molly Smith. The British Columbia setting makes this series unique. This is the first book I've read by author Vicki Delany, but I am already looking for more books in this series to read.

Table 5.12: Example of user-written reviews from the **Goodreads** dataset (for reviews exceeding 6 lines, sentences from different parts of the review are selected to fit within the table).

Chapter 6

SIRUP System

Contents

6.1	Architecture and Implementation	100
6.1.1	Profile Construction	100
6.1.2	Candidate Filtering	100
6.1.3	Personalized Ranking	101
6.1.4	Data	101
6.2	SIRUP Platform	101
6.2.1	Constructing User Profiles	102
6.2.2	Defining Search Contexts	105
6.2.3	Recommendation Results	108
6.3	Related Work	108
6.4	Conclusion	109

The previous chapter (Chapter 5) introduced review-based profiling methods for search-based recommendation, demonstrating their effectiveness through offline evaluation. Specifically, due to the absence of gold-standard query-user-item triples, we simulated an evaluation scenario for search-based recommendation. While this approach provides quantitative insights, it does not fully capture how users interact with search-based recommendation in practice.

This chapter presents our system SIRUP (**S**earch-based **I**nteractive **R**ecommendations with **U**ser **P**rofiles). SIRUP enables exploration and qualitative comparison of review-based search-based recommendation. Our interactive system allows users to define situational context by specifying a *genre*, a *given item*, their full *user profile*, or a newly formulated *query*. The platform facilitates exploration across two large book datasets, offering various methods for constructing concise user profiles.

The SIRUP system is available at <https://sirup.mpi-inf.mpg.de/>.

6.1 Architecture and Implementation

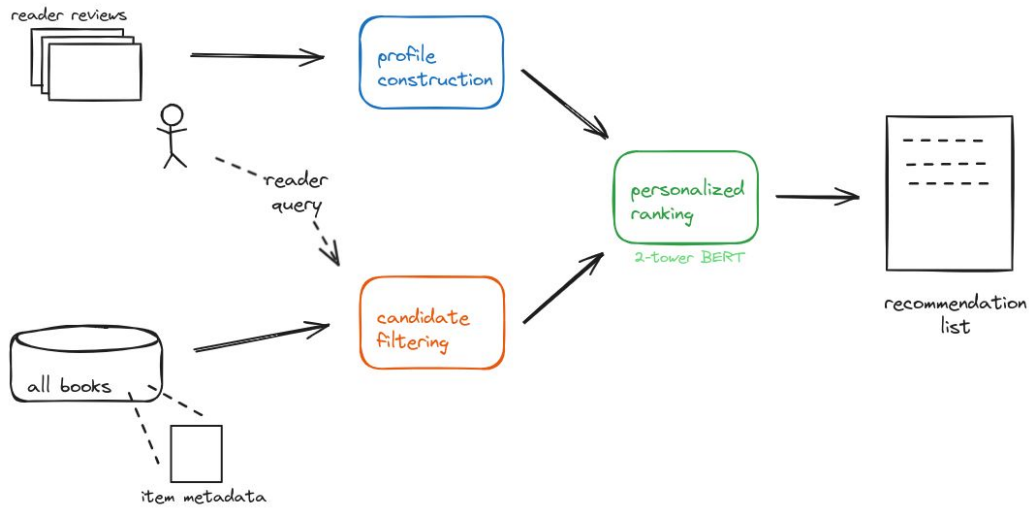


Figure 6.1: SIRUP System Architecture.

Figure 6.1 illustrates the high-level architecture of SIRUP, which consists of three primary components: profile construction, candidate filtering, and personalized ranking. The system takes as input a reader’s review history and a situational query to generate a personalized recommendation list. Our system is implemented in Python, using the Flask framework¹.

6.1.1 Profile Construction

This component constructs a concise user profile from the reader’s reviews, selecting relevant textual information that captures their preferences (see Subsection 5.2.1 of Chapter 5). In our setup, the size of the user profile is constrained to 128 tokens.

6.1.2 Candidate Filtering

In addition to the long-term reader profile, which tailors predictions to the reader’s taste, SIRUP allows users to provide situational context, reflecting their current information need. Given a reader query (e.g., a specified genre, a particular book, or a free-text search query), the system retrieves a set of candidate books from the full dataset. We employ the BM25 algorithm to obtain the top-100 candidate items, based on lexical match scores between the provided context (e.g., query text) and the book metadata, including title, genre, and description.

When selecting *existing users* from the dataset, we ensure that their books used for training are excluded from the candidate items.

¹<https://flask.palletsprojects.com/>

6.1.3 Personalized Ranking

Candidate books, filtered by the query when applicable, are ranked using a *two-tower* architecture with a BERT encoder, which scores items based on their relevance to the reader’s profile. The final ranked list is returned as the recommendation output.

The latent representations for readers’ profiles and items are generated separately by the two towers. The relevance score of an item for a given reader is computed by calculating the dot product between their respective representations. BERT-based rankers are fine-tuned on our datasets, for each user profiling configuration (see Subsection 5.2.2 of Chapter 5).

6.1.4 Data

SIRUP uses two datasets of book reviews: *Amazon (AM)* and *Goodreads (GR)*, both from the UCSD repository [McAuley, 2022]. To train the two-tower recommender model, we selected 1k text-rich users based on average review length, along with all items associated with them (resulting in 16k and 45k items for AM and GR, respectively).

To increase the selection of items for recommendation, we sampled the most reviewed items, expanding both datasets to approximately 100k items each. Given the strongly skewed distribution in popularity, most items are completely unseen during training. This presents a challenging scenario for a recommender system, emphasizing the importance of having informative user profiles.

6.2 SIRUP Platform

In this section, we describe the SIRUP platform, available at <https://sirup.mpi-inf.mpg.de/>. Users can switch between two datasets, Amazon and Goodreads, via the main navigation bar. They can also view dataset statistics, such as genre distributions, displayed in the upper section of the webpage (see Figure 6.2).

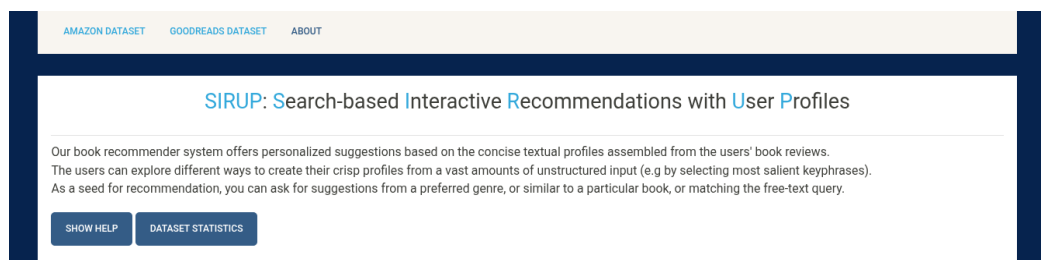


Figure 6.2: Navigation bar, help, and dataset statistics in the SIRUP platform.

The lower section of the webpage, as shown in 6.3, displays the various functionalities of our system. On the left side, users can choose between exploring *existing readers*

(predefined users) or entering as a *new reader* (manual user). On the right side, users can optionally define the situational context. Finally, after pressing the “predict” button, users can view the list of recommended items below the context-defining menu. These modules are all detailed in Subsections 6.2.1, 6.2.2, and 6.2.3.

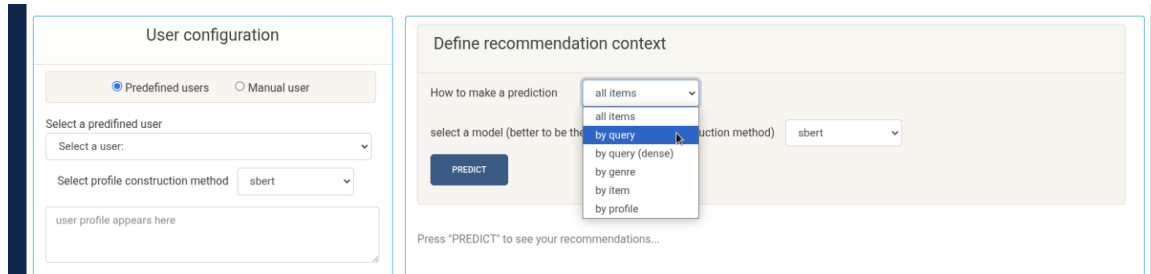


Figure 6.3: User Selection and Context Definition Modules.

6.2.1 Constructing User Profiles

The reader’s profile, based on her history of book reviews, is used to personalize the ranking of recommendations. In our system, such a profile can be defined either by selecting an existing reader from the original datasets (from a drop-down menu), or by manually creating a customized profile for a new reader.

Existing Readers: Figure 6.4 shows the setup for selecting an existing reader, where each reader in the drop-down menu is characterized by a one-line punchy description. These descriptions, generated by ChatGPT from all reviews of a reader, help to get a feel for the selected reader’s interests and tastes.

For each reader, the system also shows how much they have read: we construct three user groups: *novice* (< 5 books for AM and < 13 books for GR), *fan* (between 5 and 20 books for AM, between 13 and 71 books for GR), and *ambassador* (with more than 20 or 71 books, respectively).

For instance, reader Zoe (in Figure 6.4), who reviewed the books “Selling To The Point: Because The Information Age Demands a New Way to Sell” and “Never Split the Difference: Negotiating As If Your Life Depended On It”, is described by ChatGPT as “*B2B sales superhero: sells with storytelling; barterers like a boss*”. In total Zoe has read 3 books and is considered a *novice* in the platform. We take Zoe as our running example throughout this section.

To distill all reader’s reviews into a concise profile and eliminate useless content (e.g., Zoe’s emotional phrases, like “...an awesome ‘must read’ ...”), the system offers several methods for profile construction, as described in Subsection 5.2.1 of Chapter 5. In our example we pick the SBERT method, which selects review sentences that are semantically closest to the descriptions of the books themselves (provided by the

Select a predefined user

B2B sales superhero: sells with storytelling; barters like a boss. (novice)

Select profile construction method

sbert

[optional] extend the profile here

This is a selling book as well as a closing book. If you deal with customers, a boss, a spouse, a vendor... you need to read this book. Jeffrey Lipsius uses simple storytelling to explain why the buyer is now in charge of sales. I highly recommend it if you are in B2B sales. The case studies and writing style make it an easy and fast read. I recommend the book for B2B inside and field sales folks who are selling products to purchasing reps or SaaS to mid - level buyers. I highly recommend it if you are in B2B sales.

All user reviews (train set)

Title: Never Split the Difference: Negotiating As If Your Life Depended On It

Genres: Books, Business, Money, Management, Leadership

Review: An awesome "must read". An awesome "must read". If you deal with customers, a boss, a spouse, a vendor... you need to read this book. It's fun and entertaining. The case studies and writing style make it an easy and fast read. Yet the stories are absolutely essential. I've used it already and it works!

Title: The Lost Art of Closing: Winning the Ten Commitments That Drive Sales

Genres: Books, Business, Money, Marketing, Sales

Review: I highly recommend it if you are in B2B sales. This is a selling book as well as a closing book. Tony's writing is engaging and lively so it's a fast read. I highly recommend it if you are in

Figure 6.4: Profile Construction for Existing Readers.

AM and GR platforms). For comparison, Figure 6.5 shows Zoe’s abstractive-profile generated by Llama and keyphrase-profile generated using ChatGPT.

B2B sales, selling, closing, customer relationships, case studies, storytelling, sales strategies, buyer - centric approach, internal confidence, internal choice, internal clarity, SaaS solutions, sales training, business development, sales techniques, relationship building.

buyer - centric selling, storytelling, understanding customer needs, 10 new laws of selling, internal confidence and clarity, relevance for saas solutions, applying selling approach to personal life, engaging and lively writing, essential case studies, practical and effective.,

(a) Llama-generated abstractive profile.

(b) ChatGPT-generated keyphrase profile.

Figure 6.5: Profile examples.

New Reader: In the manual profile mode, a user has two options to create the profile (Figure 6.6): i) typing a concise text about the user’s interests and taste, or ii) entering books that the user has read and liked, providing short review texts for them.

In the latter option the user can select existing items by title (using auto-complete suggestions). When all books and reviews are entered, the user can trigger the construction of the concise profile using one of the available methods. The user can also save the entered books and reviews as a CSV file, and later load it again for a new session with the constructed profile.

☐ Predefined users
 ☒ Manual user

Please choose one of the below procedures to create a custom user profile.

☐ Option1: Input text profile
☒ Option2: Input reviews

Enter your book review history and choose one of the profile creation methods to create the profile:

Book Title	Review
Choose a book (autocomplete)	Enter the Review <input type="text"/>

You can also upload your review history from .csv file and save it (2-column csv, for book title and review)

No file chosen

(a) User may enter their profile manually.

(b) User can add their book reviews and construct a user profile using one of our methods.

Figure 6.6: New reader profile.

In both modes, the reader's profile can be edited, allowing users to modify it and add information. The text of the concise profile serves as input to the transformer-based personalized recommender model. Figure 6.7 shows an example of user edit to their automatically constructed profile. Further, in Figure 6.8, we demonstrate the changes in ranking when the two versions of user profiles are utilized.

I enjoy reading biographies of successful people.

I'm a professional looking for books on sales and marketing, specifically B2B sales. I'm interested in books that provide practical advice and strategies for success in sales. I value engaging and entertaining writing styles, as well as case studies and real - life examples. I'm looking for books that offer new perspectives and approaches to sales, such as the idea that the buyer is now in charge. I'm interested in books that focus on understanding customer needs and providing solutions that meet those needs.

I'm a professional looking for books on sales and marketing, specifically B2B sales. I'm interested in books that provide practical advice and strategies for success in sales. I value engaging and entertaining writing styles, as well as case studies and real - life examples. I'm looking for books that offer new perspectives and approaches to sales, such as the idea that the buyer is now in charge.

(a) Llama-generated abstractive profile.

(b) Modified profile.

Figure 6.7: Example of user updating their profile.

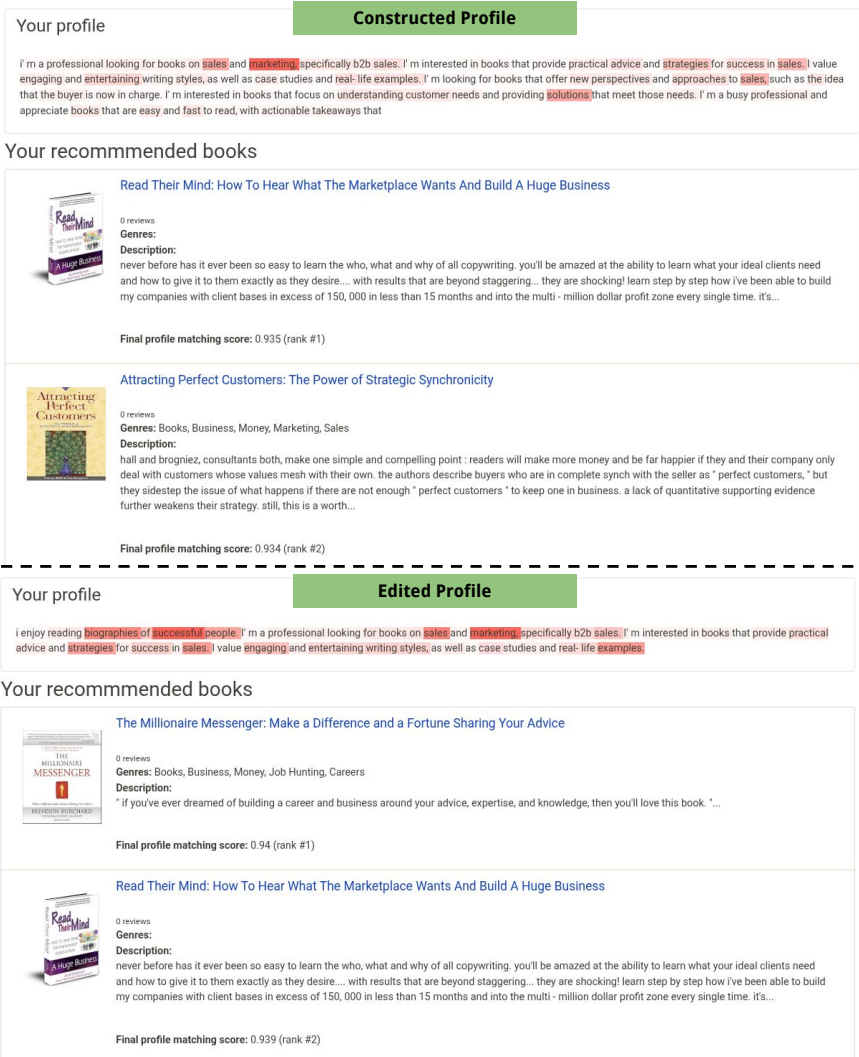


Figure 6.8: Comparison of ranked results using automatically constructed and edited user profiles.

6.2.2 Defining Search Contexts

In addition to leveraging the system-generated profiles, users can define a short-term search context, to select a matching subset of items, which are subsequently ranked by our trained transformer. The user can choose among different search modes.

By Query: In this mode, the user can specify a free-form textual query. The resulting candidate pool of top-100 results by BM25 is then re-ranked using the recommender model.

For example, Zoe queried for “romantic comedy”, getting as top recommendations some light-reading romantic novels, which connect to business topics that are expressed in Zoe’s profile (see Figure 6.9).

Define recommendation context

How to make a prediction
by_query

write a query
romantic comedy

select a model (better to be the same as text selection method)
sbert

PREDICT

Your profile

this is a **selling** book as well as a **closing** book. if you **deal** with **customers**, a boss, a spouse, a **partner**, you need to read this book. **jeffrey lipsius** **uses** simple storytelling to explain why **the buyer** is now in charge of **sales**. I highly recommend it if you are in b2b sales. **the case studies and writing style** make it an easy and fast read. I recommend the book for b2b inside and field **sales** folks who are **selling** **products** to **purchasing** reps or **saas** to mid-level buyers. I highly recommend it if you are in b2b sales.

Your recommended books

I've Got Your Number: A Novel

1 reviews

Genres: Books, Literature Fiction, Women's Fiction

Description:
a tale about how modern technology changes lives. poppy wyatt is beside herself when she loses her engagement ring and her cell phone in quick succession. when she finds a cell phone in a trash can at the hotel where she lost the ring, she seizes it, giving out the number so that people can contact her if they find the ring. it proves to be a company phone that belonged to the now former assistant of businessman...

Initial BM25 rank: #55
Final profile matching score: 0.476 (rank #1)

Shopping for a CEO (Shopping for a Billionaire series Book 7) (Volume 7)

1 reviews

Genres: Books, Romance, Contemporary

Description:
new york times and usa today bestselling author julia kent writes romantic comedy with an edge, and new adult books that push contemporary boundaries. from billionaires to bbws to rock stars...

Initial BM25 rank: #27
Final profile matching score: 0.468 (rank #2)

Define recommendation context

How to make a prediction
all items

...

Your recommended books

The One Minute Sales Person

0 reviews

Genres: Books, Business, Money, Marketing, Sales

Description:
the nameless protagonist of this slender motivational parable originally published in 1984 suffers from the existential predicament of the salesman : " the quiet fear of rejection " caused by the nagging suspicion that " the customer did not want to buy the product. " from a succession of sales gurus he learns the one minute secret - it's not selling, it's " helping people... to feel good about what they buy. " johnson, author of the business mega - seller who moved my cheese?, offers practical suggestions ranging...

Final profile matching score: 0.933 (rank #1)

How to Become a Marketing Superstar: Unexpected Rules That Ring the Cash Register

1 reviews

Genres: Books, Business, Money, Marketing, Sales

Description:
fox's fourth entry in his how to become series proves again that he has mastered the short format, advice - driven business book. the book contains 50 - odd short chapters boasting a surprising amount of useful information delivered in a street - smart style. in the chapter entitled " banish all buying barriers, " fox advises readers to eliminate anything that makes it difficult for customers to buy. about merchants featured in visa ads for not accepting amex, he says, " not accepting the american express card...

Final profile matching score: 0.931 (rank #2)

Figure 6.9: Query-based context and predictions, compared to ranking all items.

By Genre: The user can filter books by categories/genres: for Amazon the top-50 most popular genres, for Goodreads all 10 genres available in the dataset.

For instance, when Zoe selects the genre “Biographies, Memoirs” as a filter (see Figure 6.10), the system recommends biographies of business innovators, which fit well with her interests.

By Item: In the item-based mode, the user selects an item from a drop-down menu, using the entire catalog of 100K books. The user can type (prefixes of) words in book titles, and suggestions are shown by auto-completion. Top-100 books similar to the queried item would form the candidate list for personalized ranking.

By Profile: This mode uses the generated reader profile as a query to form the initial pool of candidates.

All Items: In this mode the two-tower model ranks all items in the book collection for the given reader (see Figure 6.9).

How to make a prediction

by_genre

select a genre

Biographies, Memoirs

select a model (better to be the same as text selection method)

chatgpt

PREDICT

Your profile

buyer: centric, selling, storytelling, understanding customer needs, 10 new laws of selling, internal confidence and clarity, relevance for saas solutions, applying selling approach to personal life, engaging and lively writing, essential case studies, practical and effective.,

Your recommended books

I'd Like the World to Buy a Coke: The Life and Leadership of Roberto Goizueta

1 reviews

Genres: Books, Biographies, Memoirs, Professionals, Academics

Description:
" roberto c. goizueta left a legacy that extends far beyond coca - cola. he ' s the man who showed that a company could expand by narrowing its focus to soft drink sales and that a brand name could become a marketing juggernaut " - the wall street journal (october 20, 1997), david greising (atlanta, georgia) is the atlanta bureau chief for businessweek and has been covering coca - cola and roberto goizueta...

Final profile matching score: 0.852 (rank #1)

Management Innovators: The People and Ideas that Have Shaped Modern Business

1 reviews

Genres: Books, Biographies, Memoirs, Professionals, Academics

Description:
management innovators : the people and ideas that have shaped modern business is a penetrating history of the field of management as revealed through profiles of some of its most noteworthy inventors, communicators, financiers, motivators, and gurus. management historians daniel wren and the late ronald greenwood explore this virtual who ' s who of business pioneers with an eye toward both their significant innovations and the lasting impact they have had on the corporate world. among the 31 individuals they examine are eli...

Final profile matching score: 0.831 (rank #2)

Figure 6.10: Genre-based context and predictions.

6.2.3 Recommendation Results

The results shown to the user consist of two parts (see Figures 6.9 and 6.10): i) the input profile with highlighted salient keywords identified by the model, and ii) the list of recommended books.

Weighted Input Terms: We compute the significance of the user’s input terms by utilizing BERT attention weights from the last layer. To obtain the final weights, we max-pool the weights across all attention heads and sum the resulting matrix row-wise. Note, that these weights are just an intermediate product of the two-tower model and are not directly used for matching against terms of the recommended books.

For our reader Zoe, words like “selling” and “customer” are identified among the most significant terms. Note that different text selection methods influence the choice of the important terms (ChatGPT profile in Figure 6.10 vs. SBERT sentences in Figure 6.9).

List of Book Recommendations: The list of recommendations consists of books, ranked by their similarity scores with respect to reader’s profile, calculated by the two-tower transformer. In addition to these scores, we show the books’ ranks from the initial BM25 scoring when applicable (in query-, item- or profile-based search modes). Comparing the two rankings side-by-side can provide insights on how the order of relevant books changes with personalization.

For example, for Zoe’s query (Figure 6.9), the top items after personalized re-ranking had ranks 55 and 27 by the initial candidate search with BM25.

Each item in the resulting recommendation list has metadata, including title, genres, description and the cover image of the book. For each item we specify the total number of reviews available for it; 0 reviews means that the item was *unseen* during model training.

6.3 Related Work

Most existing demonstration platforms showcase the interaction-based recommender systems [Kermany et al., 2022, Safarik et al., 2022]; in such systems the user profile is defined as a set of the items they interacted with [Safarik et al., 2022] or a set of tags [Pauw et al., 2022]. Few text-based demonstrations allow the users to enter their text in a query-like form [Arefieva et al., 2023, Petrescu et al., 2021], which is used to return top recommendations, but do not build a persistent user profile. To the best of our knowledge, we propose the first platform that recommends the items based on both long-term textual profiles and short-term situative context.

6.4 Conclusion

In this chapter, we presented the SIRUP system for text-based book recommendations, accessible at <https://sirup.mpi-inf.mpg.de>. SIRUP provides a playground for search-based recommendation, and supports exploring a suite of techniques for constructing reader profiles. The architecture of SIRUP allows straightforward transfer from books to other domains of interest.

Chapter 7

Conclusion

7.1 Summary

This thesis investigates search-based recommendation by leveraging user-generated text as a source for user profiling. User-written text presents challenges due to its inherent complexity, diversity of style, and the co-occurrence of informative cues and noisy signals. This thesis focuses on sparse long-tail data scenarios, where textual and content-based methods outperform collaborative filtering approaches. Additionally, we emphasize the importance of transparency and scrutability in the personalization process, ensuring that users can understand and control their profiles. Search-based recommendation reflects a realistic system usage scenario, where users issue situational search queries in addition to relying on their profiles and interaction history.

In Chapter 3, we introduce *questionnaire-based* user profiles, which capture explicit user preferences in short textual format. These profiles are designed to be both *concise* and *scrutable*, allowing users to inspect and edit them. We employ a suite of re-ranking approaches and additionally perform entity expansion to enrich the profiles. Our experiments demonstrate the effectiveness of such concise profiles in personalization.

In Chapter 4, we explore *user-to-user conversations* as a source for profiling. Unlike questionnaires, chats capture user preferences in an implicit manner. Using a set of re-ranking methods, we compare personalization based on two paradigms: sparse questionnaire-based profiles and rich chat-based ones. The latter have the advantage that they can be observed from the user’s natural activities without any user effort. Additionally, we examine the impact of domain-awareness in user profiling by considering both (i) the sources of data—whether generic, in-domain, or cross-domain—and (ii) domain-relevant filtering applied to each data source. Our experimental results, based on data from an extensive multi-stage user study, demonstrate that both chat-based and questionnaire-based paradigms enhance personalization effectiveness, with each excelling in different domains. Furthermore, incorporating domain-awareness in

both the profiling source and method leads to slight performance improvements.

In Chapter 5, we investigate the automatic construction of concise user profiles from *user review texts*. User reviews contain preferences in both explicit and implicit forms, and they are long, complex, and noisy. From previous chapters, we have learned that concise user profiles are effective for personalization and enhance user experience and trust in the system by being scrutable and transparent. However, creating and maintaining these profiles requires user effort over time. To retain the benefits of concise profiles without burdening the user, we explore approaches to automatically construct concise user profiles from user-written reviews. We introduce multiple techniques—ranging from extractive methods like IDF statistics to LLM-based approaches and reinforcement learning—to distill user preferences from noisy reviews. Furthermore, we explore alternative strategies for negative sampling in both evaluation and training of the recommender model, in the absence of explicit negative labels. Through extensive experiments, we observe that concise profiles are not only computationally more efficient than complete user reviews, but they also perform better due to reduced noise.

Finally, in Chapter 6, we present SIRUP, an online system for search-based interactive recommendation with user profiles. SIRUP enables users to explore and compare different profiling approaches for book recommendation, with options to add situational text queries or query-by-example. Users can also manually construct their profiles or generate them by uploading their reviews.

7.2 Outlook

The research of this thesis raises new research questions that deserve further exploration. In this section, we outline promising directions for future work.

Creating User Profiles from Heterogeneous Sources. In this thesis, we evaluated different sources for user profiling independently, focusing on their individual contributions to personalization. A natural extension of this work is to explore how multiple heterogeneous sources can be effectively integrated into a unified user profile.

The key challenge in this direction lies in addressing differences in data distribution across sources, including variations in language complexity, style, and data sparsity. For instance, a user’s structured responses in a questionnaire may emphasize explicit categorical preferences, whereas their free-form reviews could offer nuanced insights but also introduce noise. Another critical aspect is determining the relative importance of each source based on its relevance to the domain, context, and user situation.

Dynamic and Query-Aware User Profiles. An interesting future direction is leveraging dynamic user profiles instead of static ones. This calls for on-the-fly profile construction or selection of context-relevant sub-profiles from a larger static profile.

It would allow the system to adapt the profile based on the current user intent, context, and situational factors in real time. For example, when a user is searching for a fitness-related product, the system could prioritize aspects of the profile related to exercise preferences, health goals, and recent activities, while deprioritizing less relevant information, such as preferences for books or movies. Such dynamic and query-aware profiling could enhance the personalization experience, particularly as user preferences diversify and as the amount of user data increases.

Joint Training of Retriever and Recommender. This thesis leverages a two-stage retrieval and re-ranking paradigm for search-based recommendation. Rather than using a frozen retriever in the first stage and training only the recommender model (i.e., the second-stage re-ranker), an alternative approach is to jointly train both the retrieval model and the re-ranker model. Joint training has the potential to improve the system’s overall performance by aligning the retrieval and re-ranking stages. This can improve the relevance of the retrieved candidate items to the user’s query, leading to better personalization. However, a significant challenge in this approach is the lack of ground-truth data for the retrieval stage, primarily due to the absence of query-user interaction pairs in public datasets. One potential solution to this issue is query synthesis, where synthetic queries with known answers are generated to serve as pseudo-ground-truth data for training.

List of Figures

1.1	Excerpts of user-written text.	2
1.2	Search-based recommendation, for two users with different backgrounds for the same input query.	3
1.3	Examples of User Profiles.	7
2.1	An illustration of collaborative filtering (CF) showing the user-item interaction matrix and learned latent representations for users and items. The figure demonstrates that similar users and items are represented more closely in the latent space, enabling recommendation. For example, the third user is likely to be interested in the item in question (“?”) because similar users have liked it.	14
2.2	A visualization of content-based recommendation showing a user’s previous interactions in her library and their similarity to the candidate items based on matching features or content. The figure shows items similar in content to the user’s library are selected for recommendation.	16
3.1	Sparse user profile from questionnaire.	36
3.2	Examples of search results for the query “time travel” and the user’s judgements and justifications.	37
3.3	Description for movie “Requiem for a Dream”, from its Wikipedia page.	38
4.1	Excerpts from user questionnaire and chat on travel domain (with recognized named entities and concepts in boldface).	51
4.2	Example excerpts of collected chat data in different topics.	57
4.3	Complete questionnaires used for data gathering.	58
5.1	User-written review, with uninformative text crossed over. Personal background is in purple, pure sentiment in orange, most informative cues in green.	69
5.2	CUP architecture.	75
5.3	Example of search-based recommendation with text query and query-by-example.	77
5.4	Examples of item metadata from Goodreads and Amazon datasets.	80

5.5	Training time for different input lengths and trainable parameters (lines are marked every 5th epoch).	89
6.1	SIRUP System Architecture.	100
6.2	Navigation bar, help, and dataset statistics in the SIRUP platform. .	101
6.3	User Selection and Context Definition Modules.	102
6.4	Profile Construction for Existing Readers.	103
6.5	Profile examples.	103
6.6	New reader profile.	104
6.7	Example of user updating their profile.	104
6.8	Comparison of ranked results using automatically constructed and edited user profiles.	105
6.9	Query-based context and predictions, compared to ranking all items.	106
6.10	Genre-based context and predictions.	107

List of Tables

3.1	Results for different rankers when using the queries, user profiles, and user profiles enhanced by entity descriptions.	43
3.2	Experiments on user profiles without book information (Limited), and with only demographics and hobbies (Minimum) compared against the complete user profiles (Full).	44
4.1	Example queries by domain.	55
4.2	Per-user statistics for the chat data.	56
4.3	Hyperparameters for selective entity expansion.	60
4.4	NDCG@20 for different rankers and user models. Best results per row are in boldface. Statistically significant improvements over the Query-Only baselines are marked with an asterisk.	61
4.5	NDCG@20 for LM-based ranker with domain-specific vocabularies.	63
4.6	NDCG@20 for LM-based ranker with entity expansion. Best results per column are in boldface. Statistically significant improvements over None baselines are marked with an asterisk.	64
5.1	Dataset statistics.	81
5.2	Standard evaluation. The best results are shown in bold , while the second-best results are <u>underlined</u>	84
5.3	Search-based evaluation. The best results are shown in bold , while the second-best results are <u>underlined</u>	85
5.4	CUP results, by user/item groups (NDCG@5 with Search-based evaluation). The best results are in bold , while the second-best results are <u>underlined</u>	86
5.5	CUP results, by user/item groups (NDCG@5 with Standard evaluation). The best results are in bold , while the second-best results are <u>underlined</u>	88
5.6	NDCG@5 for AM-10K Sparse and Dense datasets under Standard and Search-based evaluations across user/item subgroups.	90
5.7	NDCG@5 for GR-10K Sparse and Dense datasets under Standard and Search-based evaluations across user/item subgroups.	90
5.8	Statistical comparison of different kinds of user profiles	92

5.9	User profiles constructed by various methods from Amazon dataset (truncated to max 3 lines). The original user reviews are shown in Table 5.11.	95
5.10	User profiles constructed by various methods from Goodreads dataset (truncated to max 3 lines). The original user reviews are shown in Table 5.12.	96
5.11	Example of user-written reviews from the Amazon dataset (for reviews exceeding 6 lines, sentences from different parts of the review are selected to fit within the table).	97
5.12	Example of user-written reviews from the Goodreads dataset (for reviews exceeding 6 lines, sentences from different parts of the review are selected to fit within the table).	98

Bibliography

- [Adamopoulos, 2013] Adamopoulos, P. (2013). Beyond rating prediction accuracy: on new perspectives in recommender systems. In Yang, Q., King, I., Li, Q., Pu, P., and Karypis, G., editors, *Seventh ACM Conference on Recommender Systems, RecSys '13, Hong Kong, China, October 12-16, 2013*, pages 459–462. ACM.
- [Adomavicius et al., 2022] Adomavicius, G., Bauman, K., Tuzhilin, A., and Unger, M. (2022). Context-aware recommender systems: From foundations to recent developments. In Ricci, F., Rokach, L., and Shapira, B., editors, *Recommender Systems Handbook*, pages 211–250. Springer US.
- [Adomavicius and Tuzhilin, 2005] Adomavicius, G. and Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.*, 17(6):734–749.
- [Aggarwal, 2016] Aggarwal, C. C. (2016). *Recommender Systems - The Textbook*. Springer.
- [Agichtein et al., 2006] Agichtein, E., Brill, E., Dumais, S. T., and Ragno, R. (2006). Learning user interaction models for predicting web search result preferences. In Efthimiadis, E. N., Dumais, S. T., Hawking, D., and Järvelin, K., editors, *SIGIR 2006: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, Washington, USA, August 6-11, 2006*, pages 3–10. ACM.
- [Ai et al., 2017] Ai, Q., Zhang, Y., Bi, K., Chen, X., and Croft, W. B. (2017). Learning a hierarchical embedding model for personalized product search. In Kando, N., Sakai, T., Joho, H., Li, H., de Vries, A. P., and White, R. W., editors, *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*, pages 645–654. ACM.
- [Al-Shamri, 2016] Al-Shamri, M. Y. H. (2016). User profiling approaches for demographic recommender systems. *Knowl. Based Syst.*, 100:175–187.
- [Almahairi et al., 2015] Almahairi, A., Kastner, K., Cho, K., and Courville, A. C. (2015). Learning distributed representations from reviews for collaborative filtering. In Werthner, H., Zanker, M., Golbeck, J., and Semeraro, G., editors, *Proceedings of*

- the 9th ACM Conference on Recommender Systems, RecSys 2015, Vienna, Austria, September 16-20, 2015*, pages 147–154. ACM.
- [Arefieva et al., 2023] Arefieva, V., Egger, R., Schrefl, M., and Schedl, M. (2023). Travel bird: A personalized destination recommender with tourbert and airbnb experiences. In Chua, T., Lauw, H. W., Si, L., Terzi, E., and Tsaparas, P., editors, *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, WSDM 2023, Singapore, 27 February 2023 - 3 March 2023*, pages 1164–1167. ACM.
- [Bai et al., 2020] Bai, Y., Li, X., Wang, G., Zhang, C., Shang, L., Xu, J., Wang, Z., Wang, F., and Liu, Q. (2020). Sparterm: Learning term-based sparse representation for fast text retrieval. *CoRR*, abs/2010.00768.
- [Balog, 2018] Balog, K. (2018). *Entity-Oriented Search*, volume 39 of *The Information Retrieval Series*. Springer.
- [Balog and Kenter, 2019] Balog, K. and Kenter, T. (2019). Personal knowledge graphs: A research agenda. In Fang, Y., Zhang, Y., Allan, J., Balog, K., Carterette, B., and Guo, J., editors, *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval, ICTIR 2019, Santa Clara, CA, USA, October 2-5, 2019*, pages 217–220. ACM.
- [Balog et al., 2010] Balog, K., Meij, E., and de Rijke, M. (2010). Entity search: building bridges between two worlds. In Grobelnik, M., Mika, P., Tran, D. T., and Wang, H., editors, *Proceedings of the 3rd International Semantic Search Workshop, SEMSEARCH '10, Raleigh, North Carolina, USA, April 26, 2010*, pages 9:1–9:5. ACM.
- [Balog et al., 2019] Balog, K., Radlinski, F., and Arakelyan, S. (2019). Transparent, scrutable and explainable user models for personalized recommendation. In Piwowarski, B., Chevalier, M., Gaussier, É., Maarek, Y., Nie, J., and Scholer, F., editors, *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*, pages 265–274. ACM.
- [Baltrunas and Ricci, 2014] Baltrunas, L. and Ricci, F. (2014). Experimental evaluation of context-dependent collaborative filtering using item splitting. *User Model. User Adapt. Interact.*, 24(1-2):7–34.
- [Bast et al., 2016] Bast, H., Buchhold, B., and Haussmann, E. (2016). Semantic search on text and knowledge bases. *Found. Trends Inf. Retr.*, 10(2-3):119–271.
- [Bauman et al., 2017] Bauman, K., Liu, B., and Tuzhilin, A. (2017). Aspect based recommendations: Recommending items with the most valuable aspects based on user reviews. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017*, pages 717–725. ACM.

- [Bekker and Davis, 2020] Bekker, J. and Davis, J. (2020). Learning from positive and unlabeled data: a survey. *Mach. Learn.*, 109(4):719–760.
- [Belkin and Croft, 1992] Belkin, N. J. and Croft, W. B. (1992). Information filtering and information retrieval: Two sides of the same coin? *Commun. ACM*, 35(12):29–38.
- [Bennett et al., 2011] Bennett, P. N., Radlinski, F., White, R. W., and Yilmaz, E. (2011). Inferring and using location metadata to personalize web search. In Ma, W., Nie, J., Baeza-Yates, R., Chua, T., and Croft, W. B., editors, *Proceeding of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2011, Beijing, China, July 25-29, 2011*, pages 135–144. ACM.
- [Bennett et al., 2015] Bennett, P. N., Shokouhi, M., and Caruana, R. (2015). Implicit preference labels for learning highly selective personalized rankers. In Allan, J., Croft, W. B., de Vries, A. P., and Zhai, C., editors, *Proceedings of the 2015 International Conference on The Theory of Information Retrieval, ICTIR 2015, Northampton, Massachusetts, USA, September 27-30, 2015*, pages 291–300. ACM.
- [Bennett et al., 2012] Bennett, P. N., White, R. W., Chu, W., Dumais, S. T., Bailey, P., Borisyuk, F., and Cui, X. (2012). Modeling the impact of short- and long-term behavior on search personalization. In Hersh, W. R., Callan, J., Maarek, Y., and Sanderson, M., editors, *The 35th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR '12, Portland, OR, USA, August 12-16, 2012*, pages 185–194. ACM.
- [Biancalana et al., 2013] Biancalana, C., Gasparetti, F., Micarelli, A., and Sansonetti, G. (2013). Social semantic query expansion. *ACM Trans. Intell. Syst. Technol.*, 4(4):60:1–60:43.
- [Cai and de Rijke, 2016a] Cai, F. and de Rijke, M. (2016a). Selectively personalizing query auto-completion. In Perego, R., Sebastiani, F., Aslam, J. A., Ruthven, I., and Zobel, J., editors, *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, SIGIR 2016, Pisa, Italy, July 17-21, 2016*, pages 993–996. ACM.
- [Cai and de Rijke, 2016b] Cai, F. and de Rijke, M. (2016b). A survey of query auto completion in information retrieval. *Found. Trends Inf. Retr.*, 10(4):273–363.
- [Cao et al., 2020] Cao, E., Wang, D., Huang, J., and Hu, W. (2020). Open knowledge enrichment for long-tail entities. In Huang, Y., King, I., Liu, T., and van Steen, M., editors, *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, pages 384–394. ACM / IW3C2.
- [Carpineto and Romano, 2012] Carpineto, C. and Romano, G. (2012). A survey of automatic query expansion in information retrieval. *ACM Comput. Surv.*, 44(1):1:1–1:50.

- [Carterette, 2019] Carterette, B. (2019). Music Recommendation at Spotify (Tutorial Slides).
- [Chakraborty et al., 2022] Chakraborty, P., Dutta, S., and Sanyal, D. K. (2022). Personal research knowledge graphs. In Laforest, F., Troncy, R., Simperl, E., Agarwal, D., Gionis, A., Herman, I., and Médini, L., editors, *Companion of The Web Conference 2022, Virtual Event / Lyon, France, April 25 - 29, 2022*, pages 763–768. ACM.
- [Chang and Deng, 2020] Chang, Y. W. o. c. and Deng, H. (2020). *Query understanding for search engines*. The information retrieval series. Springer.
- [Chen et al., 2023] Chen, C., Ma, W., Zhang, M., Wang, C., Liu, Y., and Ma, S. (2023). Revisiting negative sampling vs. non-sampling in implicit recommendation. *ACM Trans. Inf. Syst.*, 41(1):12:1–12:25.
- [Chen et al., 2018] Chen, C., Zhang, M., Liu, Y., and Ma, S. (2018). Neural attentional rating regression with review-level explanations. In Champin, P., Gandon, F., Lalmas, M., and Ipeirotis, P. G., editors, *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018*, pages 1583–1592. ACM.
- [Chen et al., 2022] Chen, J., Lian, D., Jin, B., Zheng, K., and Chen, E. (2022). Learning recommenders for implicit feedback with importance resampling. In Laforest, F., Troncy, R., Simperl, E., Agarwal, D., Gionis, A., Herman, I., and Médini, L., editors, *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, pages 1997–2005. ACM.
- [Chen et al., 2015] Chen, L., Chen, G., and Wang, F. (2015). Recommender systems based on user reviews: the state of the art. *User Model. User Adapt. Interact.*, 25(2):99–154.
- [Cheng and Cantú-Paz, 2010] Cheng, H. and Cantú-Paz, E. (2010). Personalized click prediction in sponsored search. In Davison, B. D., Suel, T., Craswell, N., and Liu, B., editors, *Proceedings of the Third International Conference on Web Search and Web Data Mining, WSDM 2010, New York, NY, USA, February 4-6, 2010*, pages 351–360. ACM.
- [Chirita et al., 2007] Chirita, P., Firan, C. S., and Nejdl, W. (2007). Personalized query expansion for the web. In Kraaij, W., de Vries, A. P., Clarke, C. L. A., Fuhr, N., and Kando, N., editors, *SIGIR 2007: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Amsterdam, The Netherlands, July 23-27, 2007*, pages 7–14. ACM.
- [Chirita et al., 2005] Chirita, P., Nejdl, W., Paiu, R., and Kohlschütter, C. (2005). Using ODP metadata to personalize search. In Baeza-Yates, R. A., Ziviani, N., Marchionini, G., Moffat, A., and Tait, J., editors, *SIGIR 2005: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development*

- in Information Retrieval, Salvador, Brazil, August 15-19, 2005*, pages 178–185. ACM.
- [Christakopoulou et al., 2016] Christakopoulou, K., Radlinski, F., and Hofmann, K. (2016). Towards conversational recommender systems. In Krishnapuram, B., Shah, M., Smola, A. J., Aggarwal, C. C., Shen, D., and Rastogi, R., editors, *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 815–824. ACM.
- [Degemmis et al., 2007] Degemmis, M., Lops, P., and Semeraro, G. (2007). A content-collaborative recommender that exploits wordnet-based user profiles for neighborhood formation. *User Model. User Adapt. Interact.*, 17(3):217–255.
- [Dietz, 2019] Dietz, L. (2019). ENT rank: Retrieving entities for topical information needs through entity-neighbor-text relations. In Piwowarski, B., Chevalier, M., Gaussier, É., Maarek, Y., Nie, J., and Scholer, F., editors, *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*, pages 215–224. ACM.
- [Ding et al., 2019] Ding, X., Tang, J., Liu, T. X., Xu, C., Zhang, Y., Shi, F., Jiang, Q., and Shen, D. (2019). Infer implicit contexts in real-time online-to-offline recommendation. In Teredesai, A., Kumar, V., Li, Y., Rosales, R., Terzi, E., and Karypis, G., editors, *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, pages 2336–2346. ACM.
- [Dong et al., 2025] Dong, H. V., Fang, Y., and Lauw, H. W. (2025). A contrastive framework with user, item and review alignment for recommendation. In *WSDM '25*. ACM.
- [Dubey et al., 2024] Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., and et al. (2024). The llama 3 herd of models. *CoRR*, abs/2407.21783.
- [Eke et al., 2019] Eke, C. I., Norman, A. A., Shuib, L., and Nweke, H. F. (2019). A survey of user profiling: State-of-the-art, challenges, and solutions. *IEEE Access*, 7:144907–144924.
- [Esparza et al., 2010] Esparza, S. G., O’Mahony, M. P., and Smyth, B. (2010). Effective product recommendation using the real-time web. In Bramer, M., Petridis, M., and Hopgood, A., editors, *Research and Development in Intelligent Systems XXVII - Incorporating Applications and Innovations in Intelligent Systems XVIII Proceedings of AI-2010, The Thirtieth SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence, Cambridge, England, UK, 14-16 December 2010*, pages 5–18. Springer.

- [Fang et al., 2020] Fang, H., Zhang, D., Shu, Y., and Guo, G. (2020). Deep learning for sequential recommendation: Algorithms, influential factors, and evaluations. *ACM Trans. Inf. Syst.*, 39(1):10:1–10:42.
- [Formal et al., 2021] Formal, T., Piwowarski, B., and Clinchant, S. (2021). SPLADE: sparse lexical and expansion model for first stage ranking. In Diaz, F., Shah, C., Suel, T., Castells, P., Jones, R., and Sakai, T., editors, *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 2288–2292. ACM.
- [Funk, 2006] Funk, S. (2006). Netflix Update: Try This at Home. accessed on Sep 29, 2022.
- [Gauch et al., 2007] Gauch, S., Speretta, M., Chandramouli, A., and Micarelli, A. (2007). User profiles for personalized information access. In Brusilovsky, P., Kobsa, A., and Nejdl, W., editors, *The Adaptive Web, Methods and Strategies of Web Personalization*, volume 4321 of *Lecture Notes in Computer Science*, pages 54–89. Springer.
- [Ge et al., 2018] Ge, S., Dou, Z., Jiang, Z., Nie, J., and Wen, J. (2018). Personalizing search results using hierarchical RNN with query-aware attention. In Cuzzocrea, A., Allan, J., Paton, N. W., Srivastava, D., Agrawal, R., Broder, A. Z., Zaki, M. J., Candan, K. S., Labrinidis, A., Schuster, A., and Wang, H., editors, *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, pages 347–356. ACM.
- [Geng et al., 2022] Geng, S., Liu, S., Fu, Z., Ge, Y., and Zhang, Y. (2022). Recommendation as language processing (RLP): A unified pretrain, personalized prompt & predict paradigm (P5). In Golbeck, J., Harper, F. M., Murdock, V., Ekstrand, M. D., Shapira, B., Basilico, J., Lundgaard, K. T., and Oldridge, E., editors, *RecSys '22: Sixteenth ACM Conference on Recommender Systems, Seattle, WA, USA, September 18 - 23, 2022*, pages 299–315. ACM.
- [Gerritse et al., 2022] Gerritse, E. J., Hasibi, F., and de Vries, A. P. (2022). Entity-aware transformers for entity search. In Amigó, E., Castells, P., Gonzalo, J., Carterette, B., Culpepper, J. S., and Kazai, G., editors, *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, pages 1455–1465. ACM.
- [Ghorab et al., 2013] Ghorab, M. R., Zhou, D., O'Connor, A., and Wade, V. (2013). Personalised information retrieval: survey and classification. *User Model. User Adapt. Interact.*, 23(4):381–443.
- [Gomez-Uribe and Hunt, 2016] Gomez-Uribe, C. A. and Hunt, N. (2016). The netflix recommender system: Algorithms, business value, and innovation. *ACM Trans. Manag. Inf. Syst.*, 6(4):13:1–13:19.

- [Guo et al., 2016] Guo, J., Fan, Y., Ai, Q., and Croft, W. B. (2016). A deep relevance matching model for ad-hoc retrieval. In Mukhopadhyay, S., Zhai, C., Bertino, E., Crestani, F., Mostafa, J., Tang, J., Si, L., Zhou, X., Chang, Y., Li, Y., and Sondhi, P., editors, *Proceedings of the 25th ACM International Conference on Information and Knowledge Management, CIKM 2016, Indianapolis, IN, USA, October 24-28, 2016*, pages 55–64. ACM.
- [Guo et al., 2020] Guo, J., Fan, Y., Pang, L., Yang, L., Ai, Q., Zamani, H., Wu, C., Croft, W. B., and Cheng, X. (2020). A deep look into neural ranking models for information retrieval. *Inf. Process. Manag.*, 57(6):102067.
- [Guy et al., 2010] Guy, I., Zwerdling, N., Ronen, I., Carmel, D., and Uziel, E. (2010). Social media recommendation based on people and tags. In Crestani, F., Marchand-Maillet, S., Chen, H., Efthimiadis, E. N., and Savoy, J., editors, *Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2010, Geneva, Switzerland, July 19-23, 2010*, pages 194–201. ACM.
- [He et al., 2017] He, X., Liao, L., Zhang, H., Nie, L., Hu, X., and Chua, T. (2017). Neural collaborative filtering. In Barrett, R., Cummings, R., Agichtein, E., and Gabrilovich, E., editors, *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*, pages 173–182. ACM.
- [He et al., 2016] He, X., Zhang, H., Kan, M., and Chua, T. (2016). Fast matrix factorization for online recommendation with implicit feedback. In Perego, R., Sebastiani, F., Aslam, J. A., Ruthven, I., and Zobel, J., editors, *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, SIGIR 2016, Pisa, Italy, July 17-21, 2016*, pages 549–558. ACM.
- [Hoffart et al., 2013] Hoffart, J., Suchanek, F. M., Berberich, K., and Weikum, G. (2013). YAGO2: A spatially and temporally enhanced knowledge base from wikipedia. *Artif. Intell.*, 194:28–61.
- [Hoffart et al., 2011] Hoffart, J., Yosef, M. A., Bordino, I., Fürstenau, H., Pinkal, M., Spaniol, M., Taneva, B., Thater, S., and Weikum, G. (2011). Robust disambiguation of named entities in text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 782–792. ACL.
- [Hou et al., 2024] Hou, Y., Zhang, J., Lin, Z., Lu, H., Xie, R., McAuley, J. J., and Zhao, W. X. (2024). Large language models are zero-shot rankers for recommender systems. In Goharian, N., Tonellotto, N., He, Y., Lipani, A., McDonald, G., Macdonald, C., and Ounis, I., editors, *Advances in Information Retrieval - 46th European Conference on Information Retrieval, ECIR 2024, Glasgow, UK, March*

- 24-28, 2024, *Proceedings, Part II*, volume 14609 of *Lecture Notes in Computer Science*, pages 364–381. Springer.
- [Hu et al., 2014] Hu, B., Lu, Z., Li, H., and Chen, Q. (2014). Convolutional neural network architectures for matching natural language sentences. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2042–2050.
- [Hu et al., 2019] Hu, G., Zhang, Y., and Yang, Q. (2019). Transfer meets hybrid: A synthetic approach for cross-domain collaborative filtering with text. In Liu, L., White, R. W., Mantrach, A., Silvestri, F., McAuley, J. J., Baeza-Yates, R., and Zia, L., editors, *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, pages 2822–2829. ACM.
- [Hu et al., 2008] Hu, Y., Koren, Y., and Volinsky, C. (2008). Collaborative filtering for implicit feedback datasets. In *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), December 15-19, 2008, Pisa, Italy*, pages 263–272. IEEE Computer Society.
- [Hua et al., 2023] Hua, W., Xu, S., Ge, Y., and Zhang, Y. (2023). How to index item ids for recommendation foundation models. In Ai, Q., Liu, Y., Moffat, A., Huang, X., Sakai, T., and Zobel, J., editors, *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region, SIGIR-AP 2023, Beijing, China, November 26-28, 2023*, pages 195–204. ACM.
- [Huang et al., 2024] Huang, C., Yu, T., Xie, K., Zhang, S., Yao, L., and McAuley, J. J. (2024). Foundation models for recommender systems: A survey and new perspectives. *CoRR*, abs/2402.11143.
- [Huang et al., 2007] Huang, D. W. C., Xu, Y., Trotman, A., and Geva, S. (2007). Overview of INEX 2007 link the wiki track. In Fuhr, N., Kamps, J., Lalmas, M., and Trotman, A., editors, *Focused Access to XML Documents, 6th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2007, Dagstuhl Castle, Germany, December 17-19, 2007. Selected Papers*, volume 4862 of *Lecture Notes in Computer Science*, pages 373–387. Springer.
- [Huang et al., 2013] Huang, P., He, X., Gao, J., Deng, L., Acero, A., and Heck, L. P. (2013). Learning deep structured semantic models for web search using clickthrough data. In He, Q., Iyengar, A., Nejdl, W., Pei, J., and Rastogi, R., editors, *22nd ACM International Conference on Information and Knowledge Management, CIKM'13, San Francisco, CA, USA, October 27 - November 1, 2013*, pages 2333–2338. ACM.
- [Hui et al., 2018] Hui, K., Yates, A., Berberich, K., and de Melo, G. (2018). Copacrr: A context-aware neural IR model for ad-hoc retrieval. In Chang, Y., Zhai, C., Liu, Y., and Maarek, Y., editors, *Proceedings of the Eleventh ACM International*

- Conference on Web Search and Data Mining, WSDM 2018, Marina Del Rey, CA, USA, February 5-9, 2018*, pages 279–287. ACM.
- [Jannach and Ludewig, 2017] Jannach, D. and Ludewig, M. (2017). Investigating personalized search in e-commerce. In Rus, V. and Markov, Z., editors, *Proceedings of the Thirtieth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2017, Marco Island, Florida, USA, May 22-24, 2017*, pages 645–650. AAAI Press.
- [Jannach et al., 2022] Jannach, D., Manzoor, A., Cai, W., and Chen, L. (2022). A survey on conversational recommender systems. *ACM Comput. Surv.*, 54(5):105:1–105:36.
- [Jannach and Zanker, 2024] Jannach, D. and Zanker, M. (2024). A survey on intent-aware recommender systems. *ACM Trans. Recomm. Syst.*, 3(2).
- [Jiang et al., 2020] Jiang, J., Wu, T., Roumpos, G., Cheng, H., Yi, X., Chi, E. H., Ganapathy, H., Jindal, N., Cao, P., and Wang, W. (2020). End-to-end deep attentive personalized item retrieval for online content-sharing platforms. In Huang, Y., King, I., Liu, T., and van Steen, M., editors, *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, pages 2870–2877. ACM / IW3C2.
- [Jin et al., 2020] Jin, B., Gao, C., He, X., Jin, D., and Li, Y. (2020). Multi-behavior recommendation with graph convolutional networks. In Huang, J. X., Chang, Y., Cheng, X., Kamps, J., Murdock, V., Wen, J., and Liu, Y., editors, *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 659–668. ACM.
- [Johnson et al., 2021] Johnson, J., Douze, M., and Jégou, H. (2021). Billion-scale similarity search with gpus. *IEEE Trans. Big Data*, 7(3):535–547.
- [Jones et al., 2000] Jones, K. S., Walker, S., and Robertson, S. E. (2000). A probabilistic model of information retrieval: development and comparative experiments - part 1. *Inf. Process. Manag.*, 36(6):779–808.
- [Jones, 2007] Jones, W. (2007). Personal information management. *Annu. Rev. Inf. Sci. Technol.*, 41(1):453–504.
- [Jurafsky and Martin, 2009] Jurafsky, D. and Martin, J. H. (2009). *Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition, 2nd Edition*. Prentice Hall series in artificial intelligence. Prentice Hall, Pearson Education International.
- [Kaminskas and Bridge, 2017] Kaminskas, M. and Bridge, D. (2017). Diversity, serendipity, novelty, and coverage: A survey and empirical analysis of beyond-accuracy objectives in recommender systems. *ACM Trans. Interact. Intell. Syst.*, 7(1):2:1–2:42.

- [Kang and McAuley, 2018] Kang, W. and McAuley, J. J. (2018). Self-attentive sequential recommendation. In *IEEE International Conference on Data Mining, ICDM 2018, Singapore, November 17-20, 2018*, pages 197–206. IEEE Computer Society.
- [Kang et al., 2023] Kang, W., Ni, J., Mehta, N., Sathiamoorthy, M., Hong, L., Chi, E. H., and Cheng, D. Z. (2023). Do llms understand user preferences? evaluating llms on user rating prediction. *CoRR*, abs/2305.06474.
- [Karpukhin et al., 2020] Karpukhin, V., Oguz, B., Min, S., Lewis, P. S. H., Wu, L., Edunov, S., Chen, D., and Yih, W. (2020). Dense passage retrieval for open-domain question answering. In Webber, B., Cohn, T., He, Y., and Liu, Y., editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6769–6781. Association for Computational Linguistics.
- [Kermany et al., 2022] Kermany, N. R., Yang, J., Wu, J., and Pizzato, L. (2022). Fair-srs: A fair session-based recommendation system. In Candan, K. S., Liu, H., Akoglu, L., Dong, X. L., and Tang, J., editors, *WSDM '22: The Fifteenth ACM International Conference on Web Search and Data Mining, Virtual Event / Tempe, AZ, USA, February 21 - 25, 2022*, pages 1601–1604. ACM.
- [Khattab and Zaharia, 2020] Khattab, O. and Zaharia, M. (2020). Colbert: Efficient and effective passage search via contextualized late interaction over BERT. In Huang, J. X., Chang, Y., Cheng, X., Kamps, J., Murdock, V., Wen, J., and Liu, Y., editors, *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 39–48. ACM.
- [Khusro et al., 2016] Khusro, S., Ali, Z., and Ullah, I. (2016). Recommender systems: Issues, challenges, and research opportunities. In Kim, K. J. and Joukov, N., editors, *Information Science and Applications (ICISA) 2016*, pages 1179–1189, Singapore. Springer Singapore.
- [Kobsa, 2007] Kobsa, A. (2007). Generic user modeling systems. In Brusilovsky, P., Kobsa, A., and Nejdl, W., editors, *The Adaptive Web, Methods and Strategies of Web Personalization*, volume 4321 of *Lecture Notes in Computer Science*, pages 136–154. Springer.
- [Konstan et al., 1997] Konstan, J. A., Miller, B. N., Maltz, D. A., Herlocker, J. L., Gordon, L. R., and Riedl, J. (1997). Grouplens: Applying collaborative filtering to usenet news. *Commun. ACM*, 40(3):77–87.
- [Koolen et al., 2016] Koolen, M., Bogers, T., Gäde, M., Hall, M. M., Hendrickx, I., Huurdeman, H. C., Kamps, J., Skov, M., Verberne, S., and Walsh, D. (2016). Overview of the CLEF 2016 social book search lab. In Fuhr, N., Quaresma, P., Gonçalves, T., Larsen, B., Balog, K., Macdonald, C., Cappellato, L., and Ferro,

- N., editors, *Experimental IR Meets Multilinguality, Multimodality, and Interaction - 7th International Conference of the CLEF Association, CLEF 2016, Évora, Portugal, September 5-8, 2016, Proceedings*, volume 9822 of *Lecture Notes in Computer Science*, pages 351–370. Springer.
- [Koren et al., 2009] Koren, Y., Bell, R. M., and Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37.
- [Kuzi et al., 2017] Kuzi, S., Carmel, D., Libov, A., and Raviv, A. (2017). Query expansion for email search. In Kando, N., Sakai, T., Joho, H., Li, H., de Vries, A. P., and White, R. W., editors, *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*, pages 849–852. ACM.
- [Kuzi et al., 2016] Kuzi, S., Shtok, A., and Kurland, O. (2016). Query expansion using word embeddings. In Mukhopadhyay, S., Zhai, C., Bertino, E., Crestani, F., Mostafa, J., Tang, J., Si, L., Zhou, X., Chang, Y., Li, Y., and Sondhi, P., editors, *Proceedings of the 25th ACM International Conference on Information and Knowledge Management, CIKM 2016, Indianapolis, IN, USA, October 24-28, 2016*, pages 1929–1932. ACM.
- [Kywe et al., 2012] Kywe, S. M., Lim, E., and Zhu, F. (2012). A survey of recommender systems in twitter. In Aberer, K., Flache, A., Jager, W., Liu, L., Tang, J., and Guéret, C., editors, *Social Informatics - 4th International Conference, SocInfo 2012, Lausanne, Switzerland, December 5-7, 2012. Proceedings*, volume 7710 of *Lecture Notes in Computer Science*, pages 420–433. Springer.
- [Lafferty and Zhai, 2017] Lafferty, J. D. and Zhai, C. (2017). Document language models, query models, and risk minimization for information retrieval. *SIGIR Forum*, 51(2):251–259.
- [Lalmas,] Lalmas, M. Personalizing the listening experience (invited talk), slides at <https://prs2019.splashthat.com/>.
- [Lei et al., 2020] Lei, W., He, X., de Rijke, M., and Chua, T. (2020). Conversational recommendation: Formulation, methods, and evaluation. In Huang, J. X., Chang, Y., Cheng, X., Kamps, J., Murdock, V., Wen, J., and Liu, Y., editors, *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 2425–2428. ACM.
- [Lewandowski, 2023] Lewandowski, D. (2023). *Understanding Search Engines*. Springer.
- [Li, 2011] Li, H. (2011). *Learning to Rank for Information Retrieval and Natural Language Processing*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

- [Li et al., 2019] Li, J., Jing, M., Lu, K., Zhu, L., Yang, Y., and Huang, Z. (2019). From zero-shot learning to cold-start recommendation. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 4189–4196. AAAI Press.
- [Li et al., 2010] Li, Y., Nie, J., Zhang, Y., Wang, B., Yan, B., and Weng, F. (2010). Contextual recommendation based on text mining. In Huang, C. and Jurafsky, D., editors, *COLING 2010, 23rd International Conference on Computational Linguistics, Posters Volume, 23-27 August 2010, Beijing, China*, pages 692–700. Chinese Information Processing Society of China.
- [Liang et al., 2018] Liang, D., Krishnan, R. G., Hoffman, M. D., and Jebara, T. (2018). Variational autoencoders for collaborative filtering. In Champin, P., Gandon, F., Lalmas, M., and Ipeirotis, P. G., editors, *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018*, pages 689–698. ACM.
- [Lin et al., 2024] Lin, J., Chen, B., Wang, H., Xi, Y., Qu, Y., Dai, X., Zhang, K., Tang, R., Yu, Y., and Zhang, W. (2024). Clickprompt: CTR models are strong prompt generators for adapting language models to CTR prediction. In Chua, T., Ngo, C., Kumar, R., Lauw, H. W., and Lee, R. K., editors, *Proceedings of the ACM on Web Conference 2024, WWW 2024, Singapore, May 13-17, 2024*, pages 3319–3330. ACM.
- [Lin et al., 2023] Lin, J., Dai, X., Xi, Y., Liu, W., Chen, B., Li, X., Zhu, C., Guo, H., Yu, Y., Tang, R., and Zhang, W. (2023). How can recommender systems benefit from large language models: A survey. *CoRR*, abs/2306.05817.
- [Lin et al., 2021] Lin, J., Nogueira, R. F., and Yates, A. (2021). *Pretrained Transformers for Text Ranking: BERT and Beyond*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.
- [Liu et al., 2022] Liu, B., Bai, B., Xie, W., Guo, Y., and Chen, H. (2022). Task-optimized user clustering based on mobile app usage for cold-start recommendations. In Zhang, A. and Rangwala, H., editors, *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*, pages 3347–3356. ACM.
- [Liu et al., 2019] Liu, D., Li, J., Du, B., Chang, J., and Gao, R. (2019). DAML: dual attention mutual learning between ratings and reviews for item recommendation. In Teredesai, A., Kumar, V., Li, Y., Rosales, R., Terzi, E., and Karypis, G., editors, *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, pages 344–352. ACM.

- [Liu et al., 2020a] Liu, H., Wang, Y., Peng, Q., Wu, F., Gan, L., Pan, L., and Jiao, P. (2020a). Hybrid neural recommendation with joint deep representation learning of ratings and reviews. *Neurocomputing*, 374:77–85.
- [Liu et al., 2010] Liu, J., Dolan, P., and Pedersen, E. R. (2010). Personalized news recommendation based on click behavior. In Rich, C., Yang, Q., Cavazza, M., and Zhou, M. X., editors, *Proceedings of the 15th International Conference on Intelligent User Interfaces, IUI 2010, Hong Kong, China, February 7-10, 2010*, pages 31–40. ACM.
- [Liu et al., 2020b] Liu, J., Liu, C., and Belkin, N. J. (2020b). Personalization in text information retrieval: A survey. *J. Assoc. Inf. Sci. Technol.*, 71(3):349–369.
- [Lu et al., 2018] Lu, Y., Dong, R., and Smyth, B. (2018). Coevolutionary recommendation model: Mutual learning between ratings and reviews. In Champin, P., Gandon, F., Lalmas, M., and Ipeirotis, P. G., editors, *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018*, pages 773–782. ACM.
- [Luo et al., 2023] Luo, S., Ma, C., Xiao, Y., and Song, L. (2023). Improving long-tail item recommendation with graph augmentation. In Frommholz, I., Hopfgartner, F., Lee, M., Oakes, M., Lalmas, M., Zhang, M., and Santos, R. L. T., editors, *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM 2023, Birmingham, United Kingdom, October 21-25, 2023*, pages 1707–1716. ACM.
- [Manning et al., 2008] Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to information retrieval*. Cambridge University Press.
- [Maron and Kuhns, 1960] Maron, M. E. and Kuhns, J. L. (1960). On relevance, probabilistic indexing and information retrieval. *J. ACM*, 7(3):216–244.
- [Matthijs and Radlinski, 2011] Matthijs, N. and Radlinski, F. (2011). Personalizing web search using long term browsing history. In King, I., Nejdl, W., and Li, H., editors, *Proceedings of the Forth International Conference on Web Search and Web Data Mining, WSDM 2011, Hong Kong, China, February 9-12, 2011*, pages 25–34. ACM.
- [McAuley, 2022] McAuley, J. (2022). Recommender Systems and Personalization Datasets. accessed on Sep 29, 2022.
- [McAuley and Leskovec, 2013] McAuley, J. J. and Leskovec, J. (2013). Hidden factors and hidden topics: understanding rating dimensions with review text. In Yang, Q., King, I., Li, Q., Pu, P., and Karypis, G., editors, *Seventh ACM Conference on Recommender Systems, RecSys '13, Hong Kong, China, October 12-16, 2013*, pages 165–172. ACM.

- [Menk et al., 2019] Menk, A., Sebastia, L., and Ferreira, R. (2019). Recommendation systems for tourism based on social networks: A survey. *CoRR*, abs/1903.12099.
- [Metzler et al., 2021] Metzler, D., Tay, Y., Bahri, D., and Najork, M. (2021). Rethinking search: making domain experts out of dilettantes. *SIGIR Forum*, 55(1):13:1–13:27.
- [Micarelli and Sciarrone, 2004] Micarelli, A. and Sciarrone, F. (2004). Anatomy and empirical evaluation of an adaptive web-based information filtering system. *User Model. User Adapt. Interact.*, 14(2-3):159–200.
- [Middleton et al., 2004] Middleton, S. E., Shadbolt, N., and Roure, D. D. (2004). Ontological user profiling in recommender systems. *ACM Trans. Inf. Syst.*, 22(1):54–88.
- [Mikolov et al., 2013] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Burges, C. J. C., Bottou, L., Ghahramani, Z., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 3111–3119.
- [Miller et al., 1999] Miller, D. R. H., Leek, T., and Schwartz, R. M. (1999). A hidden markov model information retrieval system. In Gey, F. C., Hearst, M. A., and Tong, R. M., editors, *SIGIR '99: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 15-19, 1999, Berkeley, CA, USA*, pages 214–221. ACM.
- [Mitra and Craswell, 2018] Mitra, B. and Craswell, N. (2018). An introduction to neural information retrieval. *Found. Trends Inf. Retr.*, 13(1):1–126.
- [Montoya et al., 2018] Montoya, D., Tanon, T. P., Abiteboul, S., Senellart, P., and Suchanek, F. M. (2018). A knowledge base for personal information management. In Berners-Lee, T., Capadisli, S., Dietze, S., Hogan, A., Janowicz, K., and Lehmann, J., editors, *Workshop on Linked Data on the Web co-located with The Web Conference 2018, LDOW@WWW 2018, Lyon, France April 23rd, 2018*, volume 2073 of *CEUR Workshop Proceedings*. CEUR-WS.org.
- [Musto et al., 2020a] Musto, C., Polignano, M., Semeraro, G., de Gemmis, M., and Lops, P. (2020a). Myrror: a platform for holistic user modeling. *User Model. User Adapt. Interact.*, 30(3):477–511.
- [Musto et al., 2020b] Musto, C., Trattner, C., Starke, A., and Semeraro, G. (2020b). Towards a knowledge-aware food recommender system exploiting holistic user models. In Kuflik, T., Torre, I., Burke, R., and Gena, C., editors, *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization, UMAP 2020, Genoa, Italy, July 12-18, 2020*, pages 333–337. ACM.

- [Mysore et al., 2023] Mysore, S., Jasim, M., McCallum, A., and Zamani, H. (2023). Editable user profiles for controllable text recommendations. In Chen, H., Duh, W. E., Huang, H., Kato, M. P., Mothe, J., and Poblete, B., editors, *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*, pages 993–1003. ACM.
- [Nadai et al., 2024] Nadai, M. D., Fabbri, F., Gigioli, P., Wang, A., Li, A., Silvestri, F., Kim, L., Lin, S., Radosavljevic, V., Ghael, S., Nyhan, D., Bouchard, H., Lalmas, M., and Damianou, A. (2024). Personalized audiobook recommendations at spotify through graph neural networks. In Chua, T., Ngo, C., Lee, R. K., Kumar, R., and Lauw, H. W., editors, *Companion Proceedings of the ACM on Web Conference 2024, WWW 2024, Singapore, Singapore, May 13-17, 2024*, pages 403–412. ACM.
- [Najork, 2023] Najork, M. (2023). Generative information retrieval. In Chen, H., Duh, W. E., Huang, H., Kato, M. P., Mothe, J., and Poblete, B., editors, *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*, page 1. ACM.
- [Ni et al., 2019] Ni, J., Li, J., and McAuley, J. J. (2019). Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In Inui, K., Jiang, J., Ng, V., and Wan, X., editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 188–197. Association for Computational Linguistics.
- [Ni et al., 2018] Ni, Y., Ou, D., Liu, S., Li, X., Ou, W., Zeng, A., and Si, L. (2018). Perceive your users in depth: Learning universal user representations from multiple e-commerce tasks. In Guo, Y. and Farooq, F., editors, *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, pages 596–605. ACM.
- [Nikolakopoulos and Karypis, 2019] Nikolakopoulos, A. N. and Karypis, G. (2019). Recwalk: Nearly uncoupled random walks for top-n recommendation. In Culpepper, J. S., Moffat, A., Bennett, P. N., and Lerman, K., editors, *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM 2019, Melbourne, VIC, Australia, February 11-15, 2019*, pages 150–158. ACM.
- [Nogueira and Cho, 2019] Nogueira, R. F. and Cho, K. (2019). Passage re-ranking with BERT. *CoRR*, abs/1901.04085.
- [Nogueira et al., 2020] Nogueira, R. F., Jiang, Z., Pradeep, R., and Lin, J. (2020). Document ranking with a pretrained sequence-to-sequence model. In Cohn, T., He, Y., and Liu, Y., editors, *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 708–718. Association for Computational Linguistics.

- [OpenAI, 2024] OpenAI (2024). Chatgpt: A large language model. Accessed: 2025-02-19.
- [Palangi et al., 2016] Palangi, H., Deng, L., Shen, Y., Gao, J., He, X., Chen, J., Song, X., and Ward, R. K. (2016). Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. *IEEE ACM Trans. Audio Speech Lang. Process.*, 24(4):694–707.
- [Paradarami et al., 2017] Paradarami, T. K., Bastian, N. D., and Wightman, J. L. (2017). A hybrid recommender system using artificial neural networks. *Expert Syst. Appl.*, 83:300–313.
- [Pauw et al., 2022] Pauw, J. D., Ruymbeek, K., and Goethals, B. (2022). Who do you think I am? interactive user modelling with item metadata. In Golbeck, J., Harper, F. M., Murdock, V., Ekstrand, M. D., Shapira, B., Basilico, J., Lundgaard, K. T., and Oldridge, E., editors, *RecSys '22: Sixteenth ACM Conference on Recommender Systems, Seattle, WA, USA, September 18 - 23, 2022*, pages 640–643. ACM.
- [Pei et al., 2019] Pei, C., Zhang, Y., Zhang, Y., Sun, F., Lin, X., Sun, H., Wu, J., Jiang, P., Ge, J., Ou, W., and Pei, D. (2019). Personalized re-ranking for recommendation. In Bogers, T., Said, A., Brusilovsky, P., and Tikk, D., editors, *Proceedings of the 13th ACM Conference on Recommender Systems, RecSys 2019, Copenhagen, Denmark, September 16-20, 2019*, pages 3–11. ACM.
- [Peña et al., 2020] Peña, F. J., O'Reilly-Morgan, D., Tragos, E. Z., Hurley, N., Duriakova, E., Smyth, B., and Lawlor, A. (2020). Combining rating and review data by initializing latent factor models with topic models for top-n recommendation. In Santos, R. L. T., Marinho, L. B., Daly, E. M., Chen, L., Falk, K., Koenigstein, N., and de Moura, E. S., editors, *RecSys 2020: Fourteenth ACM Conference on Recommender Systems, Virtual Event, Brazil, September 22-26, 2020*, pages 438–443. ACM.
- [Penha and Hauff, 2020] Penha, G. and Hauff, C. (2020). What does BERT know about books, movies and music? probing BERT for conversational recommendation. In Santos, R. L. T., Marinho, L. B., Daly, E. M., Chen, L., Falk, K., Koenigstein, N., and de Moura, E. S., editors, *RecSys 2020: Fourteenth ACM Conference on Recommender Systems, Virtual Event, Brazil, September 22-26, 2020*, pages 388–397. ACM.
- [Penha et al., 2024] Penha, G., Vardasbi, A., Palumbo, E., Nadai, M. D., and Bouchard, H. (2024). Bridging search and recommendation in generative retrieval: Does one task help the other? In Noia, T. D., Lops, P., Joachims, T., Verbert, K., Castells, P., Dong, Z., and London, B., editors, *Proceedings of the 18th ACM Conference on Recommender Systems, RecSys 2024, Bari, Italy, October 14-18, 2024*, pages 340–349. ACM.

- [Petrescu et al., 2021] Petrescu, D. A., Antognini, D., and Faltings, B. (2021). Multi-step critiquing user interface for recommender systems. In Pampín, H. J. C., Larson, M. A., Willemsen, M. C., Konstan, J. A., McAuley, J. J., Garcia-Gathright, J., Huurnink, B., and Oldridge, E., editors, *RecSys '21: Fifteenth ACM Conference on Recommender Systems, Amsterdam, The Netherlands, 27 September 2021 - 1 October 2021*, pages 760–763. ACM.
- [Ponte and Croft, 1998] Ponte, J. M. and Croft, W. B. (1998). A language modeling approach to information retrieval. In Croft, W. B., Moffat, A., van Rijsbergen, C. J., Wilkinson, R., and Zobel, J., editors, *SIGIR '98: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 24-28 1998, Melbourne, Australia*, pages 275–281. ACM.
- [Preotiuc-Pietro et al., 2015] Preotiuc-Pietro, D., Lampos, V., and Aletras, N. (2015). An analysis of the user occupational class through twitter content. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 1754–1764. The Association for Computer Linguistics.
- [Pugoy and Kao, 2020] Pugoy, R. A. and Kao, H. (2020). Bert-based neural collaborative filtering and fixed-length contiguous tokens explanation. In Wong, K., Knight, K., and Wu, H., editors, *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, AACL/IJCNLP 2020, Suzhou, China, December 4-7, 2020*, pages 143–153. Association for Computational Linguistics.
- [Pugoy and Kao, 2021] Pugoy, R. A. and Kao, H. (2021). Unsupervised extractive summarization-based representations for accurate and explainable collaborative filtering. In Zong, C., Xia, F., Li, W., and Navigli, R., editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 2981–2990. Association for Computational Linguistics.
- [Purificato et al., 2024] Purificato, E., Boratto, L., and Luca, E. W. D. (2024). User modeling and user profiling: A comprehensive survey. *CoRR*, abs/2402.09660.
- [Radlinski et al., 2022] Radlinski, F., Balog, K., Diaz, F., Dixon, L., and Wedin, B. (2022). On natural language user profiles for transparent and scrutable recommendation. In Amigó, E., Castells, P., Gonzalo, J., Carterette, B., Culpepper, J. S., and Kazai, G., editors, *SIGIR '22: The 45th International ACM SIGIR Conference*

- on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, pages 2863–2874. ACM.
- [Raffel et al., 2020] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.
- [Ramos et al., 2024] Ramos, J., Rahmani, H. A., Wang, X., Fu, X., and Lipani, A. (2024). Transparent and scrutable recommendations using natural language user profiles. In Ku, L., Martins, A., and Srikumar, V., editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL 2024, Bangkok, Thailand, August 11-16, 2024, pages 13971–13984. Association for Computational Linguistics.
- [Rashid et al., 2002] Rashid, A. M., Albert, I., Cosley, D., Lam, S. K., McNee, S. M., Konstan, J. A., and Riedl, J. (2002). Getting to know you: learning new user preferences in recommender systems. In Hammond, K. J., Gil, Y., and Leake, D., editors, *Proceedings of the 7th International Conference on Intelligent User Interfaces, IUI 2002, San Francisco, California, USA, January 13-16, 2002*, pages 127–134. ACM.
- [Rashid et al., 2008] Rashid, A. M., Karypis, G., and Riedl, J. (2008). Learning preferences of new users in recommender systems: an information theoretic approach. *SIGKDD Explor.*, 10(2):90–100.
- [Raziperchikolaei et al., 2021] Raziperchikolaei, R., Liang, G., and Chung, Y. (2021). Shared neural item representations for completely cold start problem. In Pampín, H. J. C., Larson, M. A., Willemsen, M. C., Konstan, J. A., McAuley, J. J., Garcia-Gathright, J., Huurnink, B., and Oldridge, E., editors, *RecSys '21: Fifteenth ACM Conference on Recommender Systems, Amsterdam, The Netherlands, 27 September 2021 - 1 October 2021*, pages 422–431. ACM.
- [Reimers and Gurevych, 2019] Reimers, N. and Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. In Inui, K., Jiang, J., Ng, V., and Wan, X., editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3980–3990. Association for Computational Linguistics.
- [Ricci et al., 2015] Ricci, F., Rokach, L., and Shapira, B., editors (2015). *Recommender Systems Handbook*. Springer.
- [Ricci et al., 2022] Ricci, F., Rokach, L., and Shapira, B., editors (2022). *Recommender Systems Handbook*. Springer US.
- [Robertson and Jones, 1976] Robertson, S. E. and Jones, K. S. (1976). Relevance weighting of search terms. *J. Am. Soc. Inf. Sci.*, 27(3):129–146.

- [Robertson and Zaragoza, 2009] Robertson, S. E. and Zaragoza, H. (2009). The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.
- [Rokach and Kisilevich, 2012] Rokach, L. and Kisilevich, S. (2012). Initial profile generation in recommender systems using pairwise comparison. *IEEE Trans. Syst. Man Cybern. Part C*, 42(6):1854–1859.
- [Sachdeva and McAuley, 2020] Sachdeva, N. and McAuley, J. J. (2020). How useful are reviews for recommendation? A critical review and potential improvements. In Huang, J. X., Chang, Y., Cheng, X., Kamps, J., Murdock, V., Wen, J., and Liu, Y., editors, *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 1845–1848. ACM.
- [Safarík et al., 2022] Safarík, J., Vancura, V., and Kordík, P. (2022). Repsys: Framework for interactive evaluation of recommender systems. In Golbeck, J., Harper, F. M., Murdock, V., Ekstrand, M. D., Shapira, B., Basilico, J., Lundgaard, K. T., and Oldridge, E., editors, *RecSys ’22: Sixteenth ACM Conference on Recommender Systems, Seattle, WA, USA, September 18 - 23, 2022*, pages 636–639. ACM.
- [Sakai, 2007] Sakai, T. (2007). Alternatives to bpref. In Kraaij, W., de Vries, A. P., Clarke, C. L. A., Fuhr, N., and Kando, N., editors, *SIGIR 2007: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Amsterdam, The Netherlands, July 23-27, 2007*, pages 71–78. ACM.
- [Salton and Buckley, 1988] Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Inf. Process. Manag.*, 24(5):513–523.
- [Salton et al., 1975] Salton, G., Wong, A., and Yang, C. (1975). A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620.
- [Sanner et al., 2023] Sanner, S., Balog, K., Radlinski, F., Wedin, B., and Dixon, L. (2023). Large language models are competitive near cold-start recommenders for language- and item-based preferences. In Zhang, J., Chen, L., Berkovsky, S., Zhang, M., Noia, T. D., Basilico, J., Pizzato, L., and Song, Y., editors, *Proceedings of the 17th ACM Conference on Recommender Systems, RecSys 2023, Singapore, Singapore, September 18-22, 2023*, pages 890–896. ACM.
- [Sarwar et al., 2000] Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. T. (2000). Application of dimensionality reduction in recommender system—a case study.
- [Sarwar et al., 2001] Sarwar, B. M., Karypis, G., Konstan, J. A., and Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In Shen, V. Y., Saito, N., Lyu, M. R., and Zurko, M. E., editors, *Proceedings of the Tenth International World Wide Web Conference, WWW 10, Hong Kong, China, May 1-5, 2001*, pages 285–295. ACM.

- [Schnabel et al., 2020] Schnabel, T., Amershi, S., Bennett, P. N., Bailey, P., and Joachims, T. (2020). The impact of more transparent interfaces on behavior in personalized recommendation. In Huang, J. X., Chang, Y., Cheng, X., Kamps, J., Murdock, V., Wen, J., and Liu, Y., editors, *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 991–1000. ACM.
- [Sedhain et al., 2015] Sedhain, S., Menon, A. K., Sanner, S., and Xie, L. (2015). Autorec: Autoencoders meet collaborative filtering. In Gangemi, A., Leonardi, S., and Panconesi, A., editors, *Proceedings of the 24th International Conference on World Wide Web Companion, WWW 2015, Florence, Italy, May 18-22, 2015 - Companion Volume*, pages 111–112. ACM.
- [Sen et al., 2009] Sen, S., Vig, J., and Riedl, J. (2009). Tagommenders: connecting users to items through tags. In Quemada, J., León, G., Maarek, Y. S., and Nejdl, W., editors, *Proceedings of the 18th International Conference on World Wide Web, WWW 2009, Madrid, Spain, April 20-24, 2009*, pages 671–680. ACM.
- [Sepiarskaia et al., 2018] Sepiarskaia, A., Kiseleva, J., Radlinski, F., and de Rijke, M. (2018). Preference elicitation as an optimization problem. In Pera, S., Ekstrand, M. D., Amatriain, X., and O'Donovan, J., editors, *Proceedings of the 12th ACM Conference on Recommender Systems, RecSys 2018, Vancouver, BC, Canada, October 2-7, 2018*, pages 172–180. ACM.
- [Shalom et al., 2019] Shalom, O. S., Uziel, G., and Kantor, A. (2019). A generative model for review-based recommendations. In Bogers, T., Said, A., Brusilovsky, P., and Tikk, D., editors, *Proceedings of the 13th ACM Conference on Recommender Systems, RecSys 2019, Copenhagen, Denmark, September 16-20, 2019*, pages 353–357. ACM.
- [Shalom et al., 2018] Shalom, O. S., Uziel, G., Karatzoglou, A., and Kantor, A. (2018). A word is worth a thousand ratings: Augmenting ratings using reviews for collaborative filtering. In Song, D., Liu, T., Sun, L., Bruza, P., Melucci, M., Sebastiani, F., and Yang, G. H., editors, *Proceedings of the 2018 ACM SIGIR International Conference on Theory of Information Retrieval, ICTIR 2018, Tianjin, China, September 14-17, 2018*, pages 11–18. ACM.
- [Shani et al., 2005] Shani, G., Heckerman, D., and Brafman, R. I. (2005). An mdp-based recommender system. *J. Mach. Learn. Res.*, 6:1265–1295.
- [Shen et al., 2005] Shen, X., Tan, B., and Zhai, C. (2005). Implicit user modeling for personalized search. In Herzog, O., Schek, H., Fuhr, N., Chowdhury, A., and Teiken, W., editors, *Proceedings of the 2005 ACM CIKM International Conference on Information and Knowledge Management, Bremen, Germany, October 31 - November 5, 2005*, pages 824–831. ACM.

- [Shen et al., 2014] Shen, Y., He, X., Gao, J., Deng, L., and Mesnil, G. (2014). A latent semantic model with convolutional-pooling structure for information retrieval. In Li, J., Wang, X. S., Garofalakis, M. N., Soboroff, I., Suel, T., and Wang, M., editors, *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014, Shanghai, China, November 3-7, 2014*, pages 101–110. ACM.
- [Shokouhi, 2013] Shokouhi, M. (2013). Learning to personalize query auto-completion. In Jones, G. J. F., Sheridan, P., Kelly, D., de Rijke, M., and Sakai, T., editors, *The 36th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR '13, Dublin, Ireland - July 28 - August 01, 2013*, pages 103–112. ACM.
- [Shuai et al., 2022] Shuai, J., Zhang, K., Wu, L., Sun, P., Hong, R., Wang, M., and Li, Y. (2022). A review-aware graph contrastive learning framework for recommendation. In Amigó, E., Castells, P., Gonzalo, J., Carterette, B., Culpepper, J. S., and Kazai, G., editors, *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, pages 1283–1293. ACM.
- [Sieg et al., 2007] Sieg, A., Mobasher, B., and Burke, R. D. (2007). Web search personalization with ontological user profiles. In Silva, M. J., Laender, A. H. F., Baeza-Yates, R. A., McGuinness, D. L., Olstad, B., Olsen, Ø. H., and Falcão, A. O., editors, *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management, CIKM 2007, Lisbon, Portugal, November 6-10, 2007*, pages 525–534. ACM.
- [Silvestri, 2010] Silvestri, F. (2010). Mining query logs: Turning search usage data into knowledge. *Found. Trends Inf. Retr.*, 4(1-2):1–174.
- [Smith and Linden, 2017] Smith, B. and Linden, G. (2017). Two decades of recommender systems at amazon.com. *IEEE Internet Comput.*, 21(3):12–18.
- [Song et al., 2014] Song, Y., Wang, H., and He, X. (2014). Adapting deep ranknet for personalized search. In Carterette, B., Diaz, F., Castillo, C., and Metzler, D., editors, *Seventh ACM International Conference on Web Search and Data Mining, WSDM 2014, New York, NY, USA, February 24-28, 2014*, pages 83–92. ACM.
- [Sontag et al., 2012] Sontag, D. A., Collins-Thompson, K., Bennett, P. N., White, R. W., Dumais, S. T., and Billerbeck, B. (2012). Probabilistic models for personalizing web search. In Adar, E., Teevan, J., Agichtein, E., and Maarek, Y., editors, *Proceedings of the Fifth International Conference on Web Search and Web Data Mining, WSDM 2012, Seattle, WA, USA, February 8-12, 2012*, pages 433–442. ACM.
- [Speretta and Gauch, 2005] Speretta, M. and Gauch, S. (2005). Personalized search based on user search histories. In Skowron, A., Agrawal, R., Luck, M., Yamaguchi,

- T., Morizet-Mahoudeaux, P., Liu, J., and Zhong, N., editors, *2005 IEEE / WIC / ACM International Conference on Web Intelligence (WI 2005)*, 19-22 September 2005, Compiègne, France, pages 622–628. IEEE Computer Society.
- [Stamou and Ntoulas, 2009] Stamou, S. and Ntoulas, A. (2009). Search personalization through query and page topical analysis. *User Model. User Adapt. Interact.*, 19(1-2):5–33.
- [Steck et al., 2021] Steck, H., Baltrunas, L., Elahi, E., Liang, D., Raimond, Y., and Basilico, J. (2021). Deep learning for recommender systems: A netflix case study. *AI Mag.*, 42(3):7–18.
- [Stratigi et al., 2019] Stratigi, M., Li, X., Stefanidis, K., and Zhang, Z. (2019). Ratings vs. reviews in recommender systems: A case study on the amazon movies dataset. In Welzer, T., Eder, J., Podgorelec, V., Wrembel, R., Ivanovic, M., Gamper, J., Morzy, M., Tzouramanis, T., Darmont, J., and Latific, A. K., editors, *New Trends in Databases and Information Systems, ADBIS 2019 Short Papers, Workshops BBIGAP, QAUCA, SemBDM, SIMPDA, M2P, MADEISD, and Doctoral Consortium, Bled, Slovenia, September 8-11, 2019, Proceedings*, volume 1064 of *Communications in Computer and Information Science*, pages 68–76. Springer.
- [Su and Khoshgoftaar, 2009] Su, X. and Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Adv. Artif. Intell.*, 2009:421425:1–421425:19.
- [Suglia et al., 2017] Suglia, A., Greco, C., Musto, C., de Gemmis, M., Lops, P., and Semeraro, G. (2017). A deep architecture for content-based recommendations exploiting recurrent neural networks. In Bieliková, M., Herder, E., Cena, F., and Desmarais, M. C., editors, *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization, UMAP 2017, Bratislava, Slovakia, July 09 - 12, 2017*, pages 202–211. ACM.
- [Sun et al., 2019] Sun, F., Liu, J., Wu, J., Pei, C., Lin, X., Ou, W., and Jiang, P. (2019). Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In Zhu, W., Tao, D., Cheng, X., Cui, P., Rundensteiner, E. A., Carmel, D., He, Q., and Yu, J. X., editors, *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, pages 1441–1450. ACM.
- [Sun et al., 2015] Sun, J., Wang, G., Cheng, X., and Fu, Y. (2015). Mining affective text to improve social media item recommendation. *Inf. Process. Manag.*, 51(4):444–457.
- [Sutton et al., 1999] Sutton, R. S., McAllester, D. A., Singh, S., and Mansour, Y. (1999). Policy gradient methods for reinforcement learning with function approximation. In Solia, S. A., Leen, T. K., and Müller, K., editors, *Advances in Neural Information Processing Systems 12, [NIPS Conference, Denver, Colorado, USA, November 29 - December 4, 1999]*, pages 1057–1063. The MIT Press.

- [Tan and He, 2017] Tan, Z. and He, L. (2017). An efficient similarity measure for user-based collaborative filtering recommender systems inspired by the physical resonance principle. *IEEE Access*, 5:27211–27228.
- [Tang et al., 2013a] Tang, F., Zhang, B., Zheng, J., and Gu, Y. (2013a). Friend recommendation based on the similarity of micro-blog user model. In *2013 IEEE International Conference on Green Computing and Communications (GreenCom) and IEEE Internet of Things (iThings) and IEEE Cyber, Physical and Social Computing (CPSCoM)*, Beijing, China, August 20-23, 2013, pages 2200–2204. IEEE.
- [Tang et al., 2013b] Tang, J., Hu, X., and Liu, H. (2013b). Social recommendation: a review. *Soc. Netw. Anal. Min.*, 3(4):1113–1133.
- [Teevan et al., 2005] Teevan, J., Dumais, S. T., and Horvitz, E. (2005). Personalizing search via automated analysis of interests and activities. In Baeza-Yates, R. A., Ziviani, N., Marchionini, G., Moffat, A., and Tait, J., editors, *SIGIR 2005: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Salvador, Brazil, August 15-19, 2005*, pages 449–456. ACM.
- [Teevan et al., 2008] Teevan, J., Dumais, S. T., and Liebling, D. J. (2008). To personalize or not to personalize: modeling queries with variation in user intent. In Myaeng, S., Oard, D. W., Sebastiani, F., Chua, T., and Leong, M., editors, *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2008, Singapore, July 20-24, 2008*, pages 163–170. ACM.
- [Tigunova et al., 2019] Tigunova, A., Yates, A., Mirza, P., and Weikum, G. (2019). Listening between the lines: Learning personal attributes from conversations. In Liu, L., White, R. W., Mantrach, A., Silvestri, F., McAuley, J. J., Baeza-Yates, R., and Zia, L., editors, *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, pages 1818–1828. ACM.
- [Tigunova et al., 2020] Tigunova, A., Yates, A., Mirza, P., and Weikum, G. (2020). CHARM: inferring personal attributes from conversations. In Webber, B., Cohn, T., He, Y., and Liu, Y., editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 5391–5404. Association for Computational Linguistics.
- [Vu et al., 2017] Vu, T., Nguyen, D. Q., Johnson, M., Song, D., and Willis, A. (2017). Search personalization with embeddings. In Jose, J. M., Hauff, C., Altingövde, I. S., Song, D., Albakour, D., Watt, S. N. K., and Tait, J., editors, *Advances in Information Retrieval - 39th European Conference on IR Research, ECIR 2017, Aberdeen, UK, April 8-13, 2017, Proceedings*, volume 10193 of *Lecture Notes in Computer Science*, pages 598–604.

- [Wan and McAuley, 2018] Wan, M. and McAuley, J. J. (2018). Item recommendation on monotonic behavior chains. In Pera, S., Ekstrand, M. D., Amatriain, X., and O’Donovan, J., editors, *Proceedings of the 12th ACM Conference on Recommender Systems, RecSys 2018, Vancouver, BC, Canada, October 2-7, 2018*, pages 86–94. ACM.
- [Wan et al., 2016] Wan, S., Lan, Y., Guo, J., Xu, J., Pang, L., and Cheng, X. (2016). A deep architecture for semantic matching with multiple positional sentence representations. In Schuurmans, D. and Wellman, M. P., editors, *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 2835–2841. AAAI Press.
- [Wang et al., 2013] Wang, H., He, X., Chang, M., Song, Y., White, R. W., and Chu, W. (2013). Personalized ranking model adaptation for web search. In Jones, G. J. F., Sheridan, P., Kelly, D., de Rijke, M., and Sakai, T., editors, *The 36th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR ’13, Dublin, Ireland - July 28 - August 01, 2013*, pages 323–332. ACM.
- [Wang et al., 2018] Wang, H., Zhang, F., Wang, J., Zhao, M., Li, W., Xie, X., and Guo, M. (2018). Ripplenet: Propagating user preferences on the knowledge graph for recommender systems. In Cuzzocrea, A., Allan, J., Paton, N. W., Srivastava, D., Agrawal, R., Broder, A. Z., Zaki, M. J., Candan, K. S., Labrinidis, A., Schuster, A., and Wang, H., editors, *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, pages 417–426. ACM.
- [Wang and Lim, 2023] Wang, L. and Lim, E. (2023). Zero-shot next-item recommendation using large pretrained language models. *CoRR*, abs/2304.03153.
- [Wang et al., 2022] Wang, S., Cao, L., Wang, Y., Sheng, Q. Z., Orgun, M. A., and Lian, D. (2022). A survey on session-based recommender systems. *ACM Comput. Surv.*, 54(7):154:1–154:38.
- [Wang et al., 2019] Wang, S., Hu, L., Wang, Y., Cao, L., Sheng, Q. Z., and Orgun, M. A. (2019). Sequential recommender systems: Challenges, progress and prospects. In Kraus, S., editor, *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 6332–6338. ijcai.org.
- [Wang et al., 2021] Wang, X., Ounis, I., and Macdonald, C. (2021). Leveraging review properties for effective recommendation. In Leskovec, J., Grobelnik, M., Najork, M., Tang, J., and Zia, L., editors, *WWW ’21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, pages 2209–2219. ACM / IW3C2.

- [Wen et al., 2018] Wen, J., Dou, Z., and Song, R. (2018). Personalized web search. In Liu, L. and Özsu, M. T., editors, *Encyclopedia of Database Systems, Second Edition*. Springer.
- [White, 2016] White, R. W. (2016). *Personalization and Contextualization*, page 267–304. Cambridge University Press.
- [Wu et al., 2019] Wu, C., Wu, F., Ge, S., Qi, T., Huang, Y., and Xie, X. (2019). Neural news recommendation with multi-head self-attention. In Inui, K., Jiang, J., Ng, V., and Wan, X., editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 6388–6393. Association for Computational Linguistics.
- [Wu et al., 2021a] Wu, C., Wu, F., Qi, T., and Huang, Y. (2021a). Empowering news recommendation with pre-trained language models. In Diaz, F., Shah, C., Suel, T., Castells, P., Jones, R., and Sakai, T., editors, *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 1652–1656. ACM.
- [Wu et al., 2020] Wu, F., Qiao, Y., Chen, J., Wu, C., Qi, T., Lian, J., Liu, D., Xie, X., Gao, J., Wu, W., and Zhou, M. (2020). MIND: A large-scale dataset for news recommendation. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J. R., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 3597–3606. Association for Computational Linguistics.
- [Wu and Grbovic, 2020] Wu, L. and Grbovic, M. (2020). How airbnb tells you will enjoy sunset sailing in barcelona? recommendation in a two-sided travel marketplace. In Huang, J. X., Chang, Y., Cheng, X., Kamps, J., Murdock, V., Wen, J., and Liu, Y., editors, *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 2387–2396. ACM.
- [Wu et al., 2021b] Wu, L., He, X., Wang, X., Zhang, K., and Wang, M. (2021b). A survey on neural recommendation: From collaborative filtering to content and context enriched recommendation. *CoRR*, abs/2104.13030.
- [Wu et al., 2024] Wu, L., Zheng, Z., Qiu, Z., Wang, H., Gu, H., Shen, T., Qin, C., Zhu, C., Zhu, H., Liu, Q., Xiong, H., and Chen, E. (2024). A survey on large language models for recommendation. *World Wide Web (WWW)*, 27(5):60.
- [Wu et al., 2023] Wu, S., Sun, F., Zhang, W., Xie, X., and Cui, B. (2023). Graph neural networks in recommender systems: A survey. *ACM Comput. Surv.*, 55(5):97:1–97:37.
- [Xiong et al., 2017] Xiong, C., Dai, Z., Callan, J., Liu, Z., and Power, R. (2017). End-to-end neural ad-hoc ranking with kernel pooling. In Kando, N., Sakai, T., Joho,

- H., Li, H., de Vries, A. P., and White, R. W., editors, *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*, pages 55–64. ACM.
- [Xiong et al., 2021] Xiong, L., Xiong, C., Li, Y., Tang, K., Liu, J., Bennett, P. N., Ahmed, J., and Overwijk, A. (2021). Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.
- [Xue et al., 2019] Xue, F., He, X., Wang, X., Xu, J., Liu, K., and Hong, R. (2019). Deep item-based collaborative filtering for top-n recommendation. *ACM Trans. Inf. Syst.*, 37(3):33:1–33:25.
- [Yamada et al., 2018] Yamada, I., Asai, A., Shindo, H., Takeda, H., and Takefuji, Y. (2018). Wikipedia2vec: An optimized tool for learning embeddings of words and entities from wikipedia. *CoRR*, abs/1812.06280.
- [Yamada et al., 2016] Yamada, I., Shindo, H., Takeda, H., and Takefuji, Y. (2016). Joint learning of the embedding of words and entities for named entity disambiguation. In Goldberg, Y. and Riezler, S., editors, *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*, pages 250–259. ACL.
- [Yang et al., 2016] Yang, L., Guo, Q., Song, Y., Meng, S., Shokouhi, M., McDonald, K., and Croft, W. B. (2016). Modeling user interests for zero-query ranking. In Ferro, N., Crestani, F., Moens, M., Mothe, J., Silvestri, F., Nunzio, G. M. D., Hauff, C., and Silvello, G., editors, *Advances in Information Retrieval - 38th European Conference on IR Research, ECIR 2016, Padua, Italy, March 20-23, 2016. Proceedings*, volume 9626 of *Lecture Notes in Computer Science*, pages 171–184. Springer.
- [Yang et al., 2019] Yang, W., Zhang, H., and Lin, J. (2019). Simple applications of BERT for ad hoc document retrieval. *CoRR*, abs/1903.10972.
- [Yang et al., 2022] Yang, Y., Lin, J., Zhang, X., and Wang, M. (2022). PKG: A personal knowledge graph for recommendation. In Amigó, E., Castells, P., Gonzalo, J., Carterette, B., Culpepper, J. S., and Kazai, G., editors, *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, pages 3334–3338. ACM.
- [Yao et al., 2020] Yao, J., Dou, Z., and Wen, J. (2020). Employing personal word embeddings for personalized search. In Huang, J. X., Chang, Y., Cheng, X., Kamps, J., Murdock, V., Wen, J., and Liu, Y., editors, *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 1359–1368. ACM.

- [Yen et al., 2019] Yen, A., Huang, H., and Chen, H. (2019). Personal knowledge base construction from text-based lifelogs. In Piwowarski, B., Chevalier, M., Gaussier, É., Maarek, Y., Nie, J., and Scholer, F., editors, *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*, pages 185–194. ACM.
- [Yuan et al., 2021] Yuan, F., Zhang, G., Karatzoglou, A., Jose, J. M., Kong, B., and Li, Y. (2021). One person, one model, one world: Learning continual user representation without forgetting. In Diaz, F., Shah, C., Suel, T., Castells, P., Jones, R., and Sakai, T., editors, *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 696–705. ACM.
- [Zamani and Croft, 2020] Zamani, H. and Croft, W. B. (2020). Learning a joint search and recommendation model from user-item interactions. In Caverlee, J., Hu, X. B., Lalmas, M., and Wang, W., editors, *WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020*, pages 717–725. ACM.
- [Zang et al., 2023] Zang, T., Zhu, Y., Liu, H., Zhang, R., and Yu, J. (2023). A survey on cross-domain recommendation: Taxonomies, methods, and future directions. *ACM Trans. Inf. Syst.*, 41(2):42:1–42:39.
- [Zhai, 2008] Zhai, C. (2008). Statistical language models for information retrieval: A critical review. *Found. Trends Inf. Retr.*, 2(3):137–213.
- [Zhai and Lafferty, 2001] Zhai, C. and Lafferty, J. D. (2001). A study of smoothing methods for language models applied to ad hoc information retrieval. In Croft, W. B., Harper, D. J., Kraft, D. H., and Zobel, J., editors, *SIGIR 2001: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, September 9-13, 2001, New Orleans, Louisiana, USA*, pages 334–342. ACM.
- [Zhang et al., 2020] Zhang, H., Wang, S., Zhang, K., Tang, Z., Jiang, Y., Xiao, Y., Yan, W., and Yang, W. (2020). Towards personalized and semantic retrieval: An end-to-end solution for e-commerce search via embedding learning. In Huang, J. X., Chang, Y., Cheng, X., Kamps, J., Murdock, V., Wen, J., and Liu, Y., editors, *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 2407–2416. ACM.
- [Zhang et al., 2021] Zhang, Q., Li, J., Jia, Q., Wang, C., Zhu, J., Wang, Z., and He, X. (2021). UNBERT: user-news matching BERT for news recommendation. In Zhou, Z., editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pages 3356–3362. ijcai.org.

- [Zhang et al., 2013] Zhang, W., Chen, T., Wang, J., and Yu, Y. (2013). Optimizing top-n collaborative filtering via dynamic negative item sampling. In Jones, G. J. F., Sheridan, P., Kelly, D., de Rijke, M., and Sakai, T., editors, *The 36th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR '13, Dublin, Ireland - July 28 - August 01, 2013*, pages 785–788. ACM.
- [Zhang et al., 2017] Zhang, Y., Ai, Q., Chen, X., and Croft, W. B. (2017). Joint representation learning for top-n recommendation with heterogeneous information sources. In Lim, E., Winslett, M., Sanderson, M., Fu, A. W., Sun, J., Culpepper, J. S., Lo, E., Ho, J. C., Donato, D., Agrawal, R., Zheng, Y., Castillo, C., Sun, A., Tseng, V. S., and Li, C., editors, *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017, Singapore, November 06 - 10, 2017*, pages 1449–1458. ACM.
- [Zhang and Chen, 2018] Zhang, Y. and Chen, X. (2018). Explainable recommendation: A survey and new perspectives. *CoRR*, abs/1804.11192.
- [Zhao et al., 2019] Zhao, Z., Hong, L., Wei, L., Chen, J., Nath, A., Andrews, S., Kumthekar, A., Sathiamoorthy, M., Yi, X., and Chi, E. H. (2019). Recommending what video to watch next: a multitask ranking system. In Bogers, T., Said, A., Brusilovsky, P., and Tikk, D., editors, *Proceedings of the 13th ACM Conference on Recommender Systems, RecSys 2019, Copenhagen, Denmark, September 16-20, 2019*, pages 43–51. ACM.
- [Zheng et al., 2017] Zheng, L., Noroozi, V., and Yu, P. S. (2017). Joint deep modeling of users and items using reviews for recommendation. In de Rijke, M., Shokouhi, M., Tomkins, A., and Zhang, M., editors, *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, WSDM 2017, Cambridge, United Kingdom, February 6-10, 2017*, pages 425–434. ACM.
- [Zheng et al., 2016] Zheng, Y., Tang, B., Ding, W., and Zhou, H. (2016). A neural autoregressive approach to collaborative filtering. In Balcan, M. and Weinberger, K. Q., editors, *Proceedings of the 33rd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 764–773. JMLR.org.
- [Zhou et al., 2017] Zhou, D., Wu, X., Zhao, W., Lawless, S., and Liu, J. (2017). Query expansion with enriched user profiles for personalized search utilizing folksonomy data. *IEEE Trans. Knowl. Data Eng.*, 29(7):1536–1548.
- [Zhou et al., 2018] Zhou, G., Zhu, X., Song, C., Fan, Y., Zhu, H., Ma, X., Yan, Y., Jin, J., Li, H., and Gai, K. (2018). Deep interest network for click-through rate prediction. In Guo, Y. and Farooq, F., editors, *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, pages 1059–1068. ACM.

- [Zhou et al., 2020] Zhou, Y., Dou, Z., and Wen, J. (2020). Encoding history with context-aware representation learning for personalized search. In Huang, J. X., Chang, Y., Cheng, X., Kamps, J., Murdock, V., Wen, J., and Liu, Y., editors, *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 1111–1120. ACM.
- [Zhou et al., 2021] Zhou, Y., Dou, Z., Zhu, Y., and Wen, J. (2021). PSSL: self-supervised learning for personalized search with contrastive sampling. In Demartini, G., Zuccon, G., Culpepper, J. S., Huang, Z., and Tong, H., editors, *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, pages 2749–2758. ACM.
- [Zhu et al., 2023] Zhu, Y., Yuan, H., Wang, S., Liu, J., Liu, W., Deng, C., Dou, Z., and Wen, J. (2023). Large language models for information retrieval: A survey. *CoRR*, abs/2308.07107.