

Singapore Management University

Institutional Knowledge at Singapore Management University

Dissertations and Theses Collection (Open Access)

Dissertations and Theses

11-2022

Mining product textual data for recommendation explanations

LE TRUNG HOANG

Singapore Management University, thle.2017@phdcs.smu.edu.sg

Follow this and additional works at: https://ink.library.smu.edu.sg/etd_coll



Part of the [Databases and Information Systems Commons](#)

Citation

LE TRUNG HOANG. Mining product textual data for recommendation explanations. (2022). 1-146.

Available at: https://ink.library.smu.edu.sg/etd_coll/450

This PhD Dissertation is brought to you for free and open access by the Dissertations and Theses at Institutional Knowledge at Singapore Management University. It has been accepted for inclusion in Dissertations and Theses Collection (Open Access) by an authorized administrator of Institutional Knowledge at Singapore Management University. For more information, please email cherylds@smu.edu.sg.

MINING PRODUCT TEXTUAL DATA
FOR RECOMMENDATION EXPLANATIONS

Trung-Hoang Le

SINGAPORE MANAGEMENT UNIVERSITY
2022

Mining Product Textual Data for Recommendation Explanations

by
Trung-Hoang Le

*Submitted to School of Computing and Information Systems in partial fulfillment of the
requirements for the Degree of Doctor of Philosophy in Computer Science*

Dissertation Committee:

Hady W. Lauw (Supervisor/Chair)
Associate Professor of Computer Science
Singapore Management University

Zheng Baihua
Professor of Computer Science
Singapore Management University

David Lo
Professor of Computer Science
Singapore Management University

Lina Yao
Associate Professor of Computer Science
University of New South Wales

Singapore Management University
2022

Copyright © 2022 Trung-Hoang Le

I hereby declare that this dissertation is my original work
and it has been written by me in its entirety.

I have duly acknowledged all the sources of information
which have been used in this dissertation.

This PhD dissertation has also not been submitted for any degree
in any university previously

A handwritten signature in black ink, appearing to read 'Trung-Hoang Le', written in a cursive style.

Trung-Hoang Le

29 November 2022

Abstract

Recommendation explanations help to make sense of recommendations, increasing the likelihood of adoption. While they are strongly related to explainable recommendations, which seek to provide not only accurate recommendations but also accompanying explanations for those recommendations, the task of explanation can be decoupled from that of recommendation, casting the recommendation explanation as a research problem in its own right. We can categorize recommendation explanation into *integrated* and *pipeline* approaches. The former aims at a single interpretable model for both recommendation and explanation tasks. The latter produces explanation after having recommendations by another recommendation model.

We are interested in mining product textual data for recommendation explanation. Although it is an unstructured data type, textual data may come from manufacturers, sellers, and consumers. It also appears in many places, e.g., title, summary, description, review, question and answers, etc., that could be a rich source of information for recommendation explanations.

Recommendation explanations appear in many different forms such as content-based explanation [24], rules [64], topics [55], or social [67], etc. In this dissertation, we focus on diverse natural language explanation. For instance, in Chapter 3, we develop an approach synthesizing natural language explanation, i.e., a collection of sentences highlighting product aspects of interest to the target user. Previous approaches to explainable recommendations tend to rely on rigid, standardized templates, customized only via fill-in-the-blank aspect sentiments. For more flexible, literate, and varied explanations covering various aspects of interest, we develop a post-hoc recommendation explanation approach, called *Synthesizing Explanation for Explainable Recommendation* or SEER [39], that synthesizes an explanation by selecting snippets from reviews, while optimizing for representativeness and coherence. To fit the target users’ aspect preferences, SEER contextualizes the opinions based on a compatible explainable recommendation model. Evaluation on four product categories shows the efficacy of our method as opposed to baselines based on templates, review summarization, selection, and text generation.

In Chapter 4, we enhance *review-level explanation* by leveraging additional information in the form of questions and answers (QA). The challenge is in selecting a suitable review, which is customarily addressed by assessing the relative importance or “attention” of each review to the recommendation objective. The proposed framework employs QA in an attention mechanism that aligns reviews to various QAs of an item and assesses their contribution jointly to the recommendation objective. The benefits are two-fold. For one, QA aids in selecting more useful reviews. For another, QA itself could accompany a well-aligned review in an expanded form of explanation. Experiments on datasets of ten product categories showcase the efficacies of our method as compared to comparable baselines in identifying useful reviews and QAs, while maintaining parity in recommendation performance.

For another instance, most of existing explainable recommendation approaches rely on evaluative explanation, which only assess the quality of an individual item along some aspect of interest to the user. We are interested in comparative explanations, assessing a recommended item with respect to another reference item. In particular, we propose to anchor reference items on the previously adopted items in a user’s history. Not only do we aim at providing comparative explanations involving such items, but we also formulate comparative constraints involving aspect-level comparisons between the target item and the reference items. The framework, called

Comparative Explainable Recommendation or COMPAREER [40], allows us to incorporate these constraints and integrate them with recommendation objectives involving both types of subjective and objective aspect-level quality assumptions. Experiments on public datasets of several product categories showcase the efficacies of our methodology as compared to baselines at attaining better recommendation accuracy and intuitive explanations.

In Chapter 6, we introduce and tackle a novel review selection scheme which also produce novel recommendation explanations in the form of comparative sets of reviews. While choosing among several products, users may look up reviews from each product they are considering. Due to the large number of reviews of products, selecting representative reviews from one product alone is already a challenging problem. In this work, we aim to conduct review selection for multiple products simultaneously for comparative purposes. We formulate objective functions that synchronize the review selection and design efficient algorithms to optimize for the objective functions. To narrow down the potentially long list of comparison items into a shorter list of more similar items, we construct a graph representing items' similarity and design efficient algorithms to find the maximum k -subgraph including the target item. The results are validated on real world datasets on various product categories.

Publications During Enrollment

Some parts of this dissertation are the extended/modified versions of the conference papers that were published or are under review as follows.

Chapter 3:

- Le, T.-H, and Lauw, H. W. 2020. Synthesizing aspect-driven recommendation explanations from reviews. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, 2427-2434. **Distinguished Paper Award.**

Chapter 4:

- Le, T.-H, and Lauw, H. W. 2022. Question-Attentive Review-Level Recommendation Explanation. To appear in *Proceedings of the 2022 IEEE International Conference on Big Data (Big Data)*.

Chapter 5:

- Le, T.-H, and Lauw, H. W. 2021. Explainable recommendation with comparative constraints on product aspects. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining, WSDM-21*, 967-975.

Chapter 6:

- Le, T.-H, and Lauw, H. W. Selecting Comparative Sets of Reviews Across Multiple Items. In submission to a conference.

Contents

List of Figures	vi
List of Tables	viii
1 Introduction	1
1.1 Recommendation Explanation	1
1.2 Product Textual Data and Natural Language Explanation	3
1.3 Evaluative and Comparative Explanations	6
1.4 Organization and Contributions	7
2 Related Work	11
2.1 Recommendation Explanation	11
2.2 Forms of Recommendation Explanation	13
2.2.1 Listing	14
2.2.2 Chart	14
2.2.3 Figure	15
2.2.4 Miscellaneous	18
2.3 Textual Explanation	18
2.3.1 Generic Text Explanation	19
2.3.2 Template-Based Text Explanation	19
2.3.3 Review-Level Explanation	20
2.3.4 Text Summarization	21
2.3.5 Text Generation	22
2.4 Comparative Recommendation Explanations	23
I Evaluative Recommendation Explanations	25
3 Synthesizing Explanation for Explainable Recommendation	27
3.1 Problem Formulation	27
3.2 SEER Framework	29
3.2.1 Optimization Objective	30
3.2.2 Optimal Formulation via ILP	31
3.2.3 Approximation via Greedy Algorithm	33

3.3	Opinion Contextualization	34
3.4	Compatible Recommendation Models	36
3.5	Experiments	37
3.5.1	Explanation Synthesis	39
3.5.2	Opinion Contextualization	40
3.5.3	Comparison to Baselines	40
3.5.4	Qualitative Study	44
3.6	Summary	45
4	Question-Attentive Review-Level Recommendation Explanation	47
4.1	Methodology	51
4.2	Experiments	57
4.2.1	Question and Review Alignment	60
4.2.2	Review-Level Explanation	61
4.2.3	Question-Level Explanation	64
4.2.4	Rating Prediction	64
4.2.5	Case Studies	65
4.2.6	User Studies	66
4.3	Summary	69
II	Comparative Recommendation Explanations	71
5	Explainable Recommendation with Comparative Constraints on Product Aspects	73
5.1	Product Ratings over Time	75
5.2	Notation and Formulation	78
5.3	Methodology	79
5.3.1	Subjective Aspect-Level Quality	79
5.3.2	Objective Aspect-Level Quality	82
5.4	Experiment	85
5.4.1	Setup	85
5.4.2	Ranking Performance	87
5.4.3	Comparative Constraints	88
5.4.4	Incorporating Aspects in Ranking Scores	91
5.5	Comparative Explanation	93
5.5.1	Case Study	93
5.5.2	User Study	95
5.6	Summary	96
6	Selecting Comparative Sets of Reviews Across Multiple Items	97
6.1	Preliminaries	100
6.2	Comparative Review Sets Selection	102
6.2.1	Problem Formulations	102
6.2.2	Integer-Regression Algorithm	104

6.3	Core List of Comparative Items	107
6.3.1	Integer Linear Program	108
6.3.2	Greedy Algorithm	109
6.4	Experiments	110
6.4.1	Setup	110
6.4.2	Comparative Review Sets Selection	112
6.4.3	Core List of Comparative Items	113
6.4.4	Case Study	114
6.4.5	User Study	115
6.5	Summary	117
7	Conclusion	119
7.1	Summary	119
7.2	Future Research	120
7.2.1	Improving Aspect-Level Sentiment Sentences within Explanation	121
7.2.2	Modeling Subjective and Objective Aspect-Level Quality Jointly	122
	Bibliography	123

List of Figures

1.1	Integrated and pipeline approaches for recommendation explanation. (1) is integrated approach (also called <i>model-intrinsic</i>). (2) and (3) are pipeline approaches. And (2) is also called <i>model-agnostic</i> approach. Explanation model in pipeline approaches is also called <i>post-hoc</i> model.	2
1.2	An example of textual data from a product page on Amazon.com	4
1.3	An example of evaluative and comparative explanation	7
2.1	The taxonomy of recommendation explanation forms	13
2.2	Some listings related to the product iPhone SE on Amazon.com	14
2.3	A recommendation explanation for the movie “The Sixth Sense” based on relevant users [24]. The left bar chart shows a histogram of the neighbors’ ratings that show this movie has been recommended because of its high ratings from the relevant users. The right chart shows ratings from the most to the least relevant users (left to right) show that very similar users rated the movie highly.	15
2.4	Word cloud charts visualize top words in the topic distribution for aspect <i>location</i> , <i>service</i> , and <i>room</i> [88]. The left column shows the top words of the three aspects. The middle and right columns show the top words for negative and positive ratings respectively.	16
2.5	Example of using heat map chart to highlight important region on image of the recommendation product [9].	16
2.6	Explanation by highlight important reviews (pink color) and words (green color) based on the learned attention weights of HANN [12].	17
2.7	Positive and negative images that can be used for recommendation explanations [79].	17
2.8	Multimodal review generation [80]. The first line next to each photo (bold) is generated rating & text, and the second line is the ground truth.	18
3.1	Architecture of proposed framework SEER	29
3.2	ASC2V Architecture	35
4.1	A product with question and review	48
4.2	QUESTER model	52
4.3	Example explanation: Meguiar’s Sanding Pad (explanation by Top_Rated_Useful is in grey, that by QUESTER is in green)	66

4.4	Example explanation: Medela’s Breast Pump (explanation by Top_Rated_Useful is in grey, that by QUESTER is in green)	67
4.5	Example explanation: Planet Waves Guitar Rest (explanation by Top_Rated_Useful is in grey, that by QUESTER is in green)	68
4.6	Review vs Question-Answer annotation results	69
4.7	QUESTER vs Top_Rated_Useful annotation results	69
5.1	Average rating of products launched over time	76
5.2	Average rating of products since launch time	77
5.3	AUC performance of EFM and MTER while varying number of latent factors l on Electronic data	87
5.4	Example Explanations by EFM and COMPARER_{obj}	93
5.5	Example Explanations by MTER and COMPARER_{sub}	94
6.1	“Compare with similar items” on Amazon.com	99
6.2	A visualization for linear regression of a product p_i when solving COMPARESETS+106	
6.3	An example of target-oriented heaviest 3-subgraph	108
6.4	Example selected sets of reviews of a Cellphone instance	116

List of Tables

3.1	Main Notations	28
3.2	Data statistics	38
3.3	Performance ratios of SEER-Greedy to SEER-ILP (%)	38
3.4	Comparison of representative costs: ROUGE-L	39
3.5	Opinion Contextualization	40
3.6	Comparison to Baselines: Coverage	41
3.7	Comparison to Baselines: ROUGE-1 and ROUGE-L	42
3.8	Example Explanations on a Computer instance	43
3.9	Result analysis of user study	44
4.1	Main Notations	50
4.2	Data statistics	58
4.3	Performance in question and review alignment	61
4.4	Performance in Review-Level Explanation task	62
4.5	Performance in Question-Level Explanation task	63
4.6	Rating prediction performance: Mean Square Error	64
5.1	Main Notations in COMPARER	78
5.2	Data Statistics	85
5.3	Performance of EFM and COMPARER _{obj}	88
5.4	Performance of MTER and COMPARER _{sub}	89
5.5	Constraint violations analysis. Counting function $V(\cdot)$ takes all pairs and the aspect quality weights as input and reports number of pairs violating the constraint	89
5.6	Effect of Constraint Coefficient λ_d on COMPARER _{obj}	90
5.7	Effects of Constraint Coefficient λ_d on COMPARER _{sub}	91
5.8	Constraint Violations: EFM vs. COMPARER _{obj}	92
5.9	Constraint Violations: MTER vs. COMPARER _{sub}	92
5.10	Aspects in Ranking Score: α and Number of Top Aspects	92
5.11	Analysis of User Study	95
6.1	Main Notations	101
6.2	Data statistics	110
6.3	Review alignment between target item and comparative items	112
6.4	Comparison to baselines in review alignment among comparative items	113
6.5	Performance ratios over TARGETHKS _{ILP} (%)	114

6.6	Review alignment between target item and comparison items for core list of comparative items	114
6.7	Review alignment among items for core list of comparative items	115
6.8	Result analysis of user study	116

Acknowledgments

In order to accomplish this dissertation, I first and foremost would like to express my gratitude to my supervisor, Prof. Hady Wirawan Lauw. During the past five years, I have learnt from him not only the passion in doing research and great diligence but also the genuine humility and balanced lifestyle.

This dissertation would not have been possible to complete without the insightful comments and constructive suggestion from my committee members. I would like to express my deepest appreciation to Prof. Zheng Baihua, Prof. David Lo, and Prof. Lina Yao.

Besides, I would like to sincerely thank my teammates and friends, Dr. Nguyen Thanh Son, Dr. Le Duc Trong, Dr. Le Duy Dung, Dr. Aghiles Salah, Dr. Maksim Tkachenko, Dr. Truong Quoc Tuan, Chia Chong Cher, Zhang Ce, Lee Ween Jiann, Tran Nhu Thuat, Do Dinh Hieu, Darryl Ong Rong Sheng, Tran Thanh Binh, Huynh Phu Minh, Guo Jingyao, Trieu Thi Ly Ly, as well as Pham Hong Quang, and other former colleagues for the valuable time and insightful discussions during my PhD candidature.

I would like to extend my gratitude to Singapore Management University, who financially supported my PhD study via SMU PhD Full Scholarships, SMU Presidential Doctoral Fellowships award in AY2020/2021 and AY2021/2022, as well as to Singapore Data Science Consortium (SDSC) for the 2020 Dissertation Research Fellowship award.

Last but not least, my deep gratitude goes to my family and pal, who have been wholeheartedly supporting me regardless of the distance.

Chapter 1

Introduction

1.1 Recommendation Explanation

In this digital era, we are witnessing the explosion of choices. The number of choices we are being offered is often greater on the online marketplaces, necessitating the increasing use of recommender systems to help us navigate these choices. To many, searching for products and making choices are often learning experiences in their own right. Many of the products we encounter in the search process are new to us. Therefore, while recommendations may help to focus our attention and narrow our search, these recommendations may not always immediately make sense to us. This is where explanations would go a long way in persuading users to understand and accept the recommendations.

A preponderance of research efforts in recommendation system focus on improving accuracies [24]. Many are based on collaborative filtering – recommending to a user those products that another user with historically similar adoptions have adopted – via matrix factorization – decomposing rating matrix into latent factors for each user and item [33]. While such models may perform well upon backtesting, receivers of recommendations may not always comprehend why certain products are being recommended to them [96]. The latent factors that underlie these models often lack interpretability, affecting their post-deployment effectiveness.

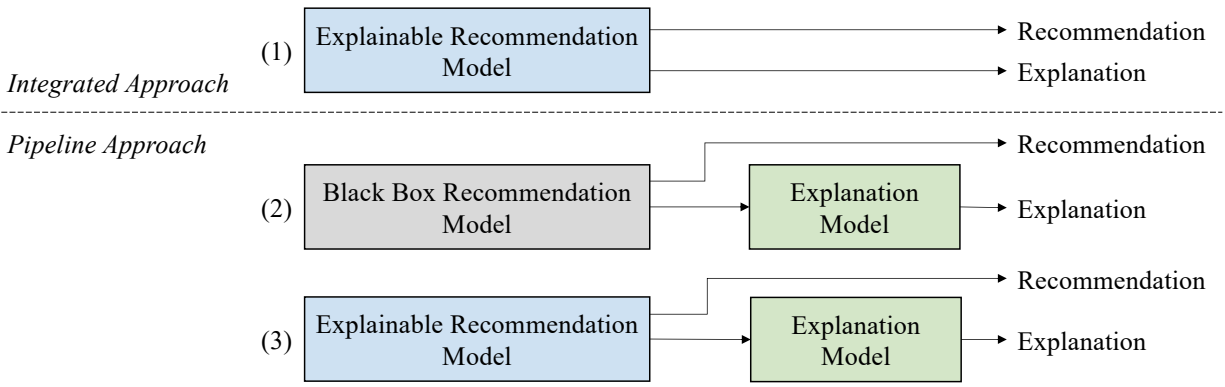


Figure 1.1: Integrated and pipeline approaches for recommendation explanation. (1) is integrated approach (also called *model-intrinsic*). (2) and (3) are pipeline approaches. And (2) is also called *model-agnostic* approach. Explanation model in pipeline approaches is also called *post-hoc* model.

Recently, there have been a surge of approaches for explainable recommendations. Explainable recommendation can be formalized as a general framework that produces not only recommendation results but also accompanying explanations that help to make sense of those recommendations regardless of the number of models being used. Both recommendation and explanation are treated as two related, yet distinct tasks.

Recommendation explanation approaches could be categorized as *integrated* and *pipeline* approaches (see Figure 1.1). The former are also called *model-intrinsic* explainable recommendation models as they aim at developing a single interpretable model for understanding how the recommendation process works (increase transparency) and interpreting the recommendation results. The latter approaches produce recommendation explanations separately. The explanation models in the pipeline approach are also called *post-hoc* model, where the explanations are generated after having the recommendation results produced by another recommendation model, which could be either a black box model or an explainable model. By this categorization, we extend the application of post-hoc explanation model upon not only black box recommendation models but also explainable recommendation models.

1.2 Product Textual Data and Natural Language Explanation

We are interested in mining product textual data for recommendation explanation. Although textual data is an unstructured data type, it is a very common form of data appearing in many places (see Figure 1.2) such as title, summary, description, specification, review, question and answers, etc., that contains descriptive information which may be used for explanations. From the manufacturers and sellers' perspective, they provide specific, factual and sometimes more emphatic information on the benefits of their products. The consumers also provide their opinion and experience on the products they have used or purchased. This increases the objectivity of the information on products. Users' past experience are also reflected in their "footprint" as they browse products online and in their feedback to the system. As a customer, we learn about a product when browsing online from information provided by the manufacturers or sellers as well as from other consumers on the product page.

In this dissertation, we aim at diverse forms of natural language explanations. Other forms of recommendation explanations will be discussed thoroughly in Section 2.2. Generic text for recommendation explanation has been used widely in many ecommerce sites, e.g., "*this item is similar to the items you viewed before*" [24] or "*people also viewed*" [77] for recommendations produced by user- or item-based collaborative filtering. Although these are brief, they increase the transparency of the model to the target user while using the recommendation system. Another approach is to list a few relevance keywords to the recommended items describing them. This is simpler than generic text, but it is descriptive enough if the extracting function extracts good quality words/phrases that well defined the recommended item. It may explain the item to the user, when he compares to his own preference. Each word/phrase in the list may have its own weight indicating the importance. In such case, we can either produce the ranked list, i.e., the most important is in front and the least one is in the end, or use word cloud with different font sizes to highlight their important, i.e., the more important word, the bigger its font size, as exemplified by [88]. However, this form of expression is not natural language as it does not



Roll over image to zoom in

Apple iPhone SE (2nd Generation), US Version, 64GB, White - Unlocked (Renewed)

Visit the Amazon Renewed Store

★★★★★ 2,241 ratings | 205 answered questions

Price: **\$289.99** + \$28.92 Shipping & Import Fees Deposit to Singapore [Details](#)

Available at a lower price from [other sellers](#) that may not offer free Prime shipping.

Size: **64GB**

64GB 128GB 256GB

Color: **White**



Service Provider: **Unlocked**

AT&T GSM Carriers T-Mobile **Unlocked** Verizon Sprint

Product grade: **Renewed**

Renewed Renewed Premium

Product works and looks like new. Backed by the 90-day Amazon Renewed Guarantee.

- This pre-owned product is not Apple certified, but has been professionally inspected, tested and cleaned by Amazon-qualified suppliers.
 - There will be no visible cosmetic imperfections when held at an arm's length.
 - This product will have a battery which exceeds 80% capacity relative to new.
 - Accessories will not be original, but will be compatible and fully functional.
- Product may come in generic Box.
- This product is eligible for a replacement or refund within 90 days of receipt if you are not satisfied. [See terms here.](#)

Product description

iPhone SE is the powerful 4.7-inch iPhone. Features A13 Bionic, one of the fastest chips in a smartphone, for incredible performance in apps, games, and photography. Portrait mode for studio-quality portraits and six lighting effects. Next-generation Smart HDR for incredible detail across highlights and shadows. Cinematic-quality 4K video. And all the advanced features of iOS. With long battery life and water resistance, it's so much of the iPhone you love, in a not so big size.

Customer questions & answers

▲
4
votes
▼

Question: [Is this the 2020 SE?](#)

Answer: Yes this is the 2020 SE, 2nd generation. A very amazing phone (Most downgrade its size) but when you buy a nice bulky case it feels amazing and also reduces your phone from being damaged.
By Kierra Prim on September 9, 2020

▲
3
votes
▼

Question: [This phone is the newest iphone se? i hope](#)

Answer: Yes.
By Jacqueline on October 13, 2020

Customer reviews



Jeffery Wilson

★★★★★ **A return to form.**

Reviewed in the United States on May 11, 2020

Color: White | Size: 64GB | **Verified Purchase**

Imagine the guts of an iPhone 11 smashed into the chassis of your favorite iPhone 5; if that sounds appealing, then this is the phone for you. Personally I loved having a non-phablet phone with this amount of power and speed. The camera is great for the price range, and call quality is also excellent. The one drawback is a shorter battery life than what I am used to.

82 people found this helpful

Figure 1.2: An example of textual data from a product page on Amazon.com

read naturally and we need some prior knowledge to understand the idea of this presentation. Another well known approach to natural language explanation that has been adopted by many existing works is to use a standardized template for explanations, substituting words within a prespecified sentence. For instance, EFM [96] has templates for positive and negative opinions, each time substituting only the *[aspect]*, e.g.,:

You might be interested in *[battery life]*, on which this product performs well.

You might be interested in *[lens]*, on which this product performs poorly.

To increase variation beyond “well”, “poorly”, MTER [85] further specifies an *<opinion phrase>*, e.g.,:

Its *[battery life]* is *<long>*.

To produce such textual recommendation explanations, both EFM [96] and MTER [85] extract aspect-level sentiment from reviews. These templates could be repetitive, robotic, and limited in their expressiveness. They tend to read less naturally than a human-created sentence.

A product review contains sentences that recount a user’s experience with the product, which often go some way towards explaining her choices *post-adoption*. Leveraging this explanatory quality, but intending to explain a predicted recommendation *pre-adoption*, other methods propose to produce textual review as explanation. We look to the literature on mining reviews for works that could potentially be adapted for this application. One possible formulation is text summarization, i.e., abstractive [43] or extractive [2]; an explanation could be a summary of reviews of a product. Another is review selection, whereby we select review(s) based on some criteria; for instance, the criterion could be the most helpfulness [8]. While it could benefit from well-formed sentences coherently laid out within a review, this approach suffers from a limited search space (whole reviews), and may end up supplying the same explanation, even if user’s preferences are different. Yet another possible approach is text generation [16], i.e., to train a natural language generation model on past reviews to construct a new review from scratch. However, this approach requires a sufficient amount of data to learn; for products with few reviews

it would tend to overfit and generate repetitive sentences. We propose a synthesis approach (described in Chapter 3) that addresses the mentioned limitation, which could be categorized into extractive text summarization approach. We also perform extensive experiments to evaluate the efficacies of our proposed method in comparing to other approaches. In Chapter 4, we leverage questions and answers (QA) as additional information to improve *review-level explanation*. By employing QA in an attention mechanism that aligns reviews to various QAs of an item, the proposed neural network jointly assesses the contribution of QAs and reviews to the recommendation objective. The experiment result showcase the efficacies of the proposed method as compared to comparable baselines in identifying useful reviews and QAs, while maintaining parity in recommendation performance.

1.3 Evaluative and Comparative Explanations

In terms of item quality, we can categorize the recommendation explanation approaches as *evaluative* and *comparative* approaches. Most of the recent approaches to recommendation explanation are *evaluative* approaches, they often explain an item assessing the quality of that individual. It could be the overall quality or the quality of the item along some aspects of interest to the user. e.g., “*You might be interested in [battery life], on which this product performs well.*”. Contrasted to *evaluative*, the *comparative* approaches assess a recommended item in comparison to another reference item [40] or multiple items in its clusters [5]. Figure 1.3 shows an example of evaluative in contrast with comparative explanation of a recommended bluetooth speaker.

A problem is to identify which item(s) to serve as reference to a recommended item. There are several reasonable options. One could be a comparable substitute under consideration, e.g., a buyer of washing machines may wish to know how other washers in the market compare to the recommended one. Another could be a previously purchased item by the target user. We propose a comparative explainable recommendation model with the latter setting (see Chapter 5). There are many websites offering user to select a pair or list of items for comparison, e.g., *versus.com*.



Figure 1.3: An example of evaluative and comparative explanation

In Chapter 6, we propose to select comparative sets of reviews for a given set of comparative items.

1.4 Organization and Contributions

This dissertation contributes novel recommendation explanation approaches. Extensive experiments on datasets of several product categories showcase the efficacies of our proposed approaches. The remaining part of this dissertation is structured as follows:

- In Chapter 2: We review related works in the literature for recommendation explanations, especially for textual explanations and comparative explanation.
- In Chapter 3: We elaborate a *post-hoc* recommendation explanation approach for explainable recommendation models such as EFM and MTER. We observed that a product review contains sentences that recount a user’s experience with the product, which often go some way towards explaining her choices *post-adoption*. Leveraging this explanatory quality, but intending to explain a predicted recommendation *pre-adoption*, we propose to “synthesize” an explanation by taking snippets from various reviews and putting them together in a coherent manner. Fitted to the recommendation, this synthesis benefits from the ex-

pressiveness of human-created review sentences, and yet is still flexible enough to produce varied explanations given the wide array of combinatorial selections from rich review corpora. Moreover, since a candidate sentence may bear in-built sentiment potentially incompatible to a user’s own, we expand candidate selection to all aspect-relevant sentences by incorporating opinion contextualization for sentiment compatibility. The proposed method is called *Synthesizing Explanation for Explainable Recommendation* or SEER. This work was published in Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI) 2020 [39] and received a Distinguished Paper Award at the conference in January 2021.

- In Chapter 4: We improve review-level recommendation explanation by leveraging additional information in the form of questions and answers (QA). The proposed framework employs QA in an attention mechanism that aligns reviews to various QAs of an item and assesses their contribution jointly to the recommendation objective. The benefits are two-fold. For one, QA aids in selecting more useful reviews. For another, QA itself could accompany a well-aligned review in an expanded form of explanation. This work has been accepted as a short paper for the 2022 IEEE International Conference on Big Data (IEEE BigData 2022).
- In Chapter 5: Most of previous approaches rely on evaluative explanations, assessing the quality of an individual item along some aspects of interest to the user. For instance, a pioneering work EFM [96] produces an explanation in the form of “*You might be interested in [aspect], on which this product performs [well/poorly].*”. In turn, another well-known model MTER [85] produces an explanation in the form of “*Its [aspect] is [opinion] phrase.*”. We posit that users are interested in choice-making, gaining information from relative comparisons and we propose to anchor reference items on the previously adopted items in a user’s history. Not only do we aim at providing comparative explanations involving such items, but we also formulate comparative constraints involving aspect-level

comparisons between the target item and the reference items. The framework is called *Comparative Explainable Recommendation* or COMPARER. This framework allows us to incorporate these constraints and integrate them with recommendation objectives involving both types of subjective and objective aspect-level quality assumptions. COMPARER seeks a comparative explanation for a recommended item, with respect to another reference item in the form “[*recommended item*] is better at [*an aspect*] than [*reference item*], but worse at [*another aspect*].”. This work was published in Proceedings of the 14th ACM International Conference on Web Search and Data Mining (WSDM) 2021 [40].

- In Chapter 6: When comparing several products online, users may wish to look up the representative reviews from each product. Due to the large number of reviews of products, selecting reviews from one product alone is a challenging problem. In this work, we aim to conduct review selection for multiple products simultaneously. We formulate objective function that synchronizes the review selection and design efficient algorithm to optimize for the objective function. To reduce the number of comparisons items, we construct item graph representing items’ similarity and design efficient algorithm to find the maximum subgraph including the target item.
- In Chapter 7: We conclude this dissertation and discuss future research directions.

Chapter 2

Related Work

2.1 Recommendation Explanation

Our main problem is recommendation explanation, which is strongly related to explainable recommendation. While recommendation helps user in narrowing down options for their choice-making process, recommendation explanation helps to make sense of recommendation result. In combining both objectives, explainable recommendation seeks to not only accurate recommendations but also accompanying explanations for those recommendations. The recommendation explanation task could be decoupled from that of recommendation to gain the benefit of generalized explanation without relying on the accurate recommendation objective. This motivates us in creating novel recommendation explanations but not limited to the explanations that have been proposed by prior explainable recommendation models, i.e., [39] proposed a novel synthesize approach for recommendation explanation to create more flexible, literate, and varied explanations than that of the base explainable recommendation models such as EFM [96] and MTER [85], which are limited in their expressiveness by using a standardized template.

Literature categorizes explainable recommendation research into *model-intrinsic* and *model-agnostic* approaches [47] by considering the explainability of either the recommendation methods or the recommendation results. The former develops model in such a way that its decision

mechanism is transparent and easy to explain, which also means the accuracy is no longer the only objective. The latter considers recommendation model be a black box and generates an explanation after the recommendation has been produced by another recommendation model. We find that the explanation model may also work upon another explainable recommendation model. To this extent, we categorize recommendation explanation into *integrated* and *pipeline* approaches (see Figure 1.1). The former approach produces both recommendation and explanation at once, identical to *model-intrinsic* approach. The latter produces them separately by different models, similar to *model-agnostic*, yet the explanation model is not tied itself only to another black box recommendation model. The explanation model in *pipeline* approaches is also called *post-hoc* model, where the explanations are generated after having the recommendations produced by another black box or explainable recommendation model.

Integrated recommendation explanation approaches include [3, 75, 96] based on matrix factorization, [6, 85] based on tensor factorization, [55, 73, 84, 88] combining matrix factorization with topic modeling. Others enhance explainable recommendation models using trees [18], graph [23], knowledge graph [72, 87], social network [67], photos [9]. Among them, there are works utilizing attention mechanism in neural networks for explanations [9, 87]. Our proposed QUESTER (see Chapter 4) and COMPAREER (see Chapter 5) are integrated recommendation approaches.

Other than models that directly design for pipeline recommendation explanation approaches [39, 64], other approaches to recommendation explanation include but are not limited to sentiment analysis [79], document summarization [2], review synthesis [60], text generation [16]. Our proposed SEER (see Chapter 3) and COMPARESETS (see Chapter 6) are post-hoc models in a pipeline approach because the recommendations are assumed to be given.

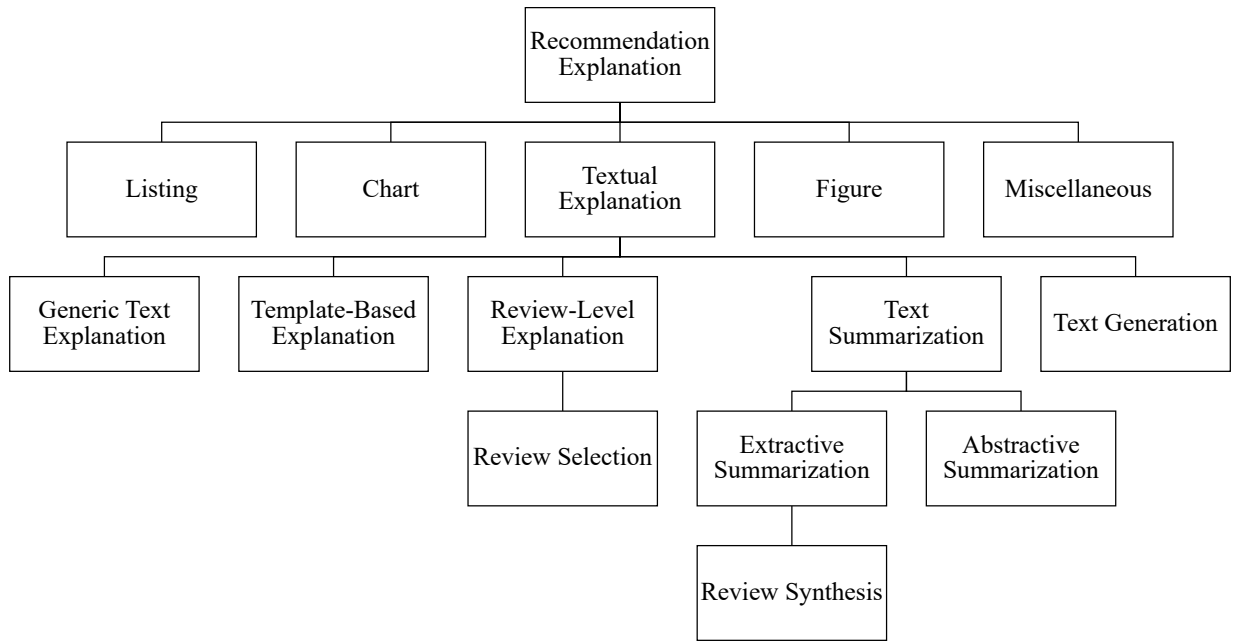


Figure 2.1: The taxonomy of recommendation explanation forms

2.2 Forms of Recommendation Explanation

Recommendation explanations may come from one or many different sources of information, i.e., model transparency, incorporating multimodality data, or using explanation models. They could be presented in various forms including text, listing, charts, and figure. Figure 2.1 shows the taxonomy of various recommendation explanation forms. In this section, we will discuss other forms of recommendation explanations, text explanation forms will be discussed in Section 2.3. Using text is easy to understand as it is natural language. Listing is simpler but sometimes difficult to elaborate. Other forms such as charts and figure may contain richer information in a more compact form. However, these forms require users to have prior knowledge to read information from them as they may contain higher level abstraction than just using natural language text.

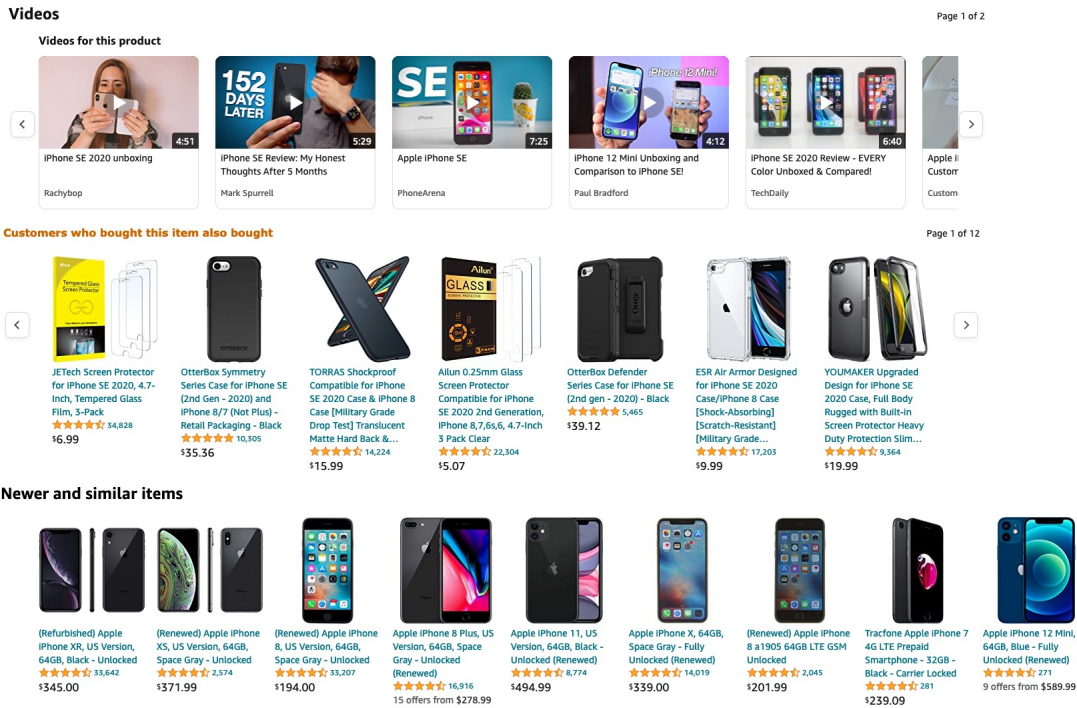


Figure 2.2: Some listings related to the product iPhone SE on Amazon.com

2.2.1 Listing

Listing is a simple form to recommendation explanation. This has been used widely in many ecommerce sites, e.g., a listing of related/frequently bought together products. This form of presentation is very useful in the case when we want to refer to many other sources as references for the recommendation item. Figure 2.2 shows the listing related to the product iPhone SE on Amazon.com. The intention is to provide further information for the user when browsing this product on the page. The user can take a look on those before making purchasing decision.

2.2.2 Chart

Chart is a powerful tool for data visualization that uses graphical representation in which data is represented by symbols, numbers, bars, lines, or slices, etc. Figure 2.3 shows explanations for a movie by presenting a bar chart of rating distribution of neighbor users and another bar chart showing neighborhood ratings base on their relatedness.

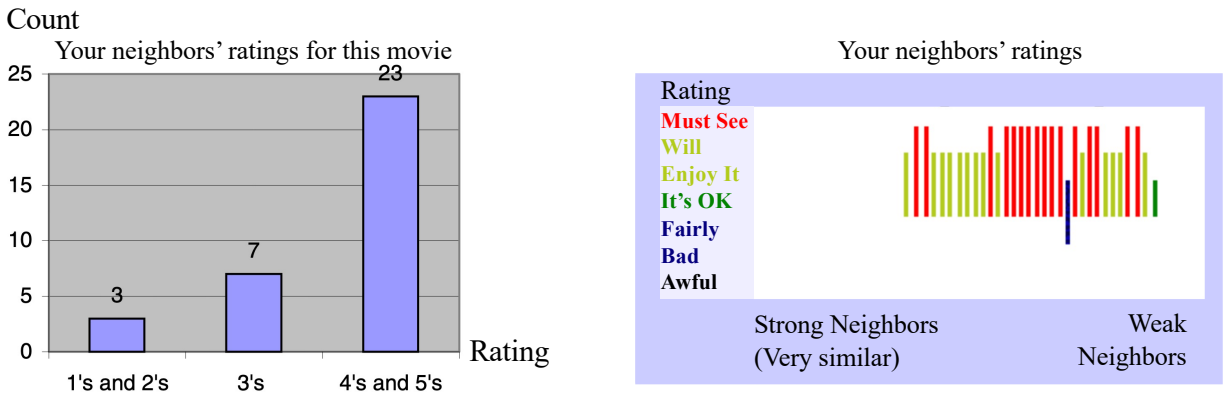


Figure 2.3: A recommendation explanation for the movie “The Sixth Sense” based on relevant users [24]. The left bar chart shows a histogram of the neighbors’ ratings that show this movie has been recommended because of its high ratings from the relevant users. The right chart shows ratings from the most to the least relevant users (left to right) show that very similar users rated the movie highly.

Another example is to use word cloud chart, which is useful to represent the distribution of words in term of their importance. The bigger the word, the more important it is. Figure 2.4 shows an example of word cloud chart explaining for hotel recommendations generated based on latent topic modeling with textual reviews [88].

In some cases, we would like to highlight the regions of the individual (e.g., photo, paragraph, etc.) that we use for explanation. [9] used the learned attention weights of the model to highlight the important parts of the image for visual explanation (see Figure 2.5). HANN [12] highlights the important reviews and words by the boldness of highlighting regions(see Figure 2.6), i.e., the bolder the region, the more important it is.

2.2.3 Figure

“A picture is worth a thousand words” is common adage saying that sometimes a single image may convey its meaning or essence more effectively than a description. For instance, [79] presents images as visual sentiment that can be used for recommendation explanation (see Figure 2.7).

Explanation Analysis of HANN

B00HG32UUV: You know, it's kind of pretty but definitely cheap. You get what you pay for, I guess. The ball at the end of the chain in the back arrived broken off but in the package. I put the bracelet on my wrist that I already had a magnetic bracelet on - turns out that THIS bracelet's not magnetic, not metal, and probably just painted plastic (Tibetan silver?) The same thing's true about the "turquoise" stone - probably just painted plastic. It's cheap but cute. I like it but probably wouldn't recommend it to anyone. You get what you pay for.

B00E1ZXX3G: Beautiful - I LOVE it! I've had a lot of compliments on it! - Very dainty, and fits nicely. Nice clasp, too!

B00BTD411E: I didn't want to get my seven-year-old grandson a digital watch, so I was excited to find this one at a great price. I gave this watch to him for Christmas, and he wears it all the time - and has learned to TELL time, too.

B0009KNC5Q: Very nice - exactly what I wanted - delicate, sterling silver, nice catch. I only wish I'd gotten another for my daughter.

Figure 2.6: Explanation by highlight important reviews (pink color) and words (green color) based on the learned attention weights of HANN [12].



Positive images of drinks, glasses



Locanda Verde

République

Bubbly Tea

Blue Frog's Local 22

Bubbly Tea

Negative images of drinks, glasses

Figure 2.7: Positive and negative images that can be used for recommendation explanations [79].



	Photo	Rating	Review
Philz Coffee		4.6	i 've had a few times for the best breakfast sandwich .
		4.0	the avocado toast was surprisingly good .
A16		4.2	i was n't sure to try the pizza .
		3.0	i think i might want to try their other pizzas which might be better tasting than their funghi .

Figure 2.8: Multimodal review generation [80]. The first line next to each photo (bold) is generated rating & text, and the second line is the ground truth.

2.2.4 Miscellaneous

We can also combine more than one forms for recommendation explanation. For instance, MRG [80] takes image as input for multimodal review generation (see Figure 2.8), in which the image can be taken from other reviewer for explanation. This model predicts rating for a pair of user and item and uses that with input image for review generation. [81] introduces visual aspect attention network to select sentences in reviews aligning with images (i.e., visual sentiment).

2.3 Textual Explanation

In this section, we will provide a comprehensive review for textual form of recommendation explanations. Many of which could still serve the purpose of recommendation explanation despite not having been designed specifically for that.

2.3.1 Generic Text Explanation

This is a traditional approach to recommendation explanation. One is to describe how the recommendation system works, helping user understand how the recommendation results being produced, which will help increase transparency of the recommendation models. User- and item-based collaborative filtering have been using generic text such as “*this item is similar to the items you viewed before*” [24] or “*people also viewed*” [77].

2.3.2 Template-Based Text Explanation

A pioneer work on template-based text explanation is EFM [96], which customizes standardized templates emphasizing positive/negative aspects in the following form

```
You might be interested in [aspect], on which this product
performs [well/poorly].
```

Each time substitutes only the *[aspect]* within the top and bottom of ranking aspect among user’s most cared aspects via their predicted item quality scores, and the word *well/poorly* will be applied for top/bottom aspect accordingly. This form of explanation is also adopted by DEAML [18], where they further incorporate *explicit aspect hierarchy* of the items and users that support explanation generation from multi-level aspects.

To increase the variation beyond “well”, “poorly”, MTER [85] further specifies *<opinion phrases>* along with the substituting aspects,

```
Its [aspect] is <opinion phrases>
```

For example, “*Its [grip] is <firmer/soft/rubbery>. Its [quality] is <sound/sturdy/smooth>. Its [cost] is <original/lower/monthly>.*” is an explanation for an Amazon product recommendation shown in the author case study.

Another work FacT [75] produces another template explanation as

```
We recommend this item to you because its [good/excellent]
[aspect] matches with your [emphasize/taste] on [aspect].
```

This template emphasizes a high quality aspect for the recommended item that matches the user preference by integrating regression trees to guide the learning of latent factor models, and uses the learnt tree structure to explain the recommendations.

We also develop a template-based text explanation, called *Comparative Explainable Recommendation* or COMPAREX [40] (see Chapter 5), which produces an explanation for a recommended item with respect to a another reference item as

```
[recommended item] is better at [an aspect] than
[reference item], but worse at [another aspect].
```

The main difference of this approach to previously discussed ones is that others produce evaluative explanations while COMPAREX produces comparative explanation. Where evaluative explanations only assess the quality of a single product in and of itself. The comparative explanation assesses the quality of the product in comparing to another item or other items.

2.3.3 Review-Level Explanation

The term *review-level explanation* has been introduced by [8]. This work obtains useful item reviews as review-level recommendation explanation via attention mechanism in their neural network architecture. In addition, HRDR [51] uses multilayer perceptron to encode user's ratings (resp. item's ratings) as user features (resp. item's features) and use that as query for attention layer to weight the contribution of each review to rating prediction. HFT [55] could select the review with the closest topic distribution to the item's topic distribution. Although this explanation is not personalized, the explanations may help user in considering the recommended item based on what other reviewers have been discussed and the useful reviews should contain useful infor-

mation about the item. We leverage questions and answers as additional information to improve review-level explanation (see Chapter 4). By using attention mechanism on QAs that aligns to reviews, QAs aids in selecting more useful reviews. QA itself could accompany a well-aligned review in an expanded form of explanation. There are various problems related to review-level recommendation explanation. One is to predict the helpfulness of online reviews [17]. There are a wide range of studies on this problem.

In a general view, review-level recommendation explanation is a review selection approach. Most of previous works focus on selecting a subset of reviews from a large collection of reviews, consider only one item at a time. [36] proposed to select a subset of reviews that represent the majority opinions on all aspects. Similarly, [82] expanded to cover both positive and negative opinions, such that selects a subset of reviews that collectively provide both the negative and positive opinions on each aspect. [37] optimizes for opinion distribution, providing a subset of reviews that present a statistically capture the proportion of opinions of an item. In addition, [89, 92] extends on the notion of review quality. We develop a novel method that formulate selecting sets of reviews for multiple products simultaneously (see Chapter 6).

There are other formulations for predicting helpful reviews or ranking reviews [15, 19, 29, 48, 49, 52, 54, 70, 74, 83, 95], without any concern about the recommendation objective.

Another formulation is to select personalized review [12, 27], which is orthogonal to selecting useful reviews. [12] uses GRU as text encoder to encode word-level and review-level representation and learn the contribution of each word/review to the rating prediction. [27] selects personalized review based on extracted aspects.

2.3.4 Text Summarization

In broader view, the reason we want to select review for recommendation explanation is that there are too many of them that may exhaust the user when trying to read them all. Another approach reduces the amount of text of a bigger corpus of reviews that is review summariza-

tion [26, 32, 59, 71, 93, 100]. Summarization approaches could be categorized as abstractive [43] or extractive [2]. The abstractive approach seeks to generate a short text for a larger corpus, which is also related to text generation (see Section 2.3.5). [43] generates tip from review and rating. The extractive approach combines sentences or snippets (i.e., phrases) as summarization. [2] combines sentences from reviews based on representativeness objective. However, without incorporating user preference into summarization, this approach cannot produce personalized recommendation explanations. We develop a synthesize framework to recommendation explanation for compatible explainable recommendation models (see Chapter 3), which is in this category, that synthesizes an explanation by assuming an aspect demand is specified as input, listing the number of sentences required for each aspect, and selecting sentences from various reviews, while optimizing for representativeness and coherence. Since a candidate sentence may bear in-built sentiment potentially incompatible to a user’s own, we expand candidate selection to all aspect-relevant sentences by incorporating opinion contextualization for sentiment compatibility. [86] applies reinforcement learning approach for selecting sentences that agree with predicting rating.

2.3.5 Text Generation

For text generation, recent works utilize recurrent neural network architecture such as LSTM with attention to generate textual review for a given item to a target user. [16] takes into account the user, item, and given rating. [61] incorporates the user and item, as well as starter phrases. [62] uses history reviews and keywords as attributes. Other methods address explainable recommendation problem and also generate textual review along side with the recommendation, which they try to predict ratings and generate reviews in multi-task learning manner [10, 21, 43, 44, 45, 80]. [43] uses the predicted rating as sentiment, along with user and item factors as context to generate explanation text. [53] extends on review textual features. [10]

conditions on concepts from an oracle¹. [80] further attends on visual aspects. [44] explicitly uses aspect keywords to generate explanation. [45] uses Transformer, a well-known language modeling technique, for personalized review generation. [21] applies transfer learning from pretrained language model for review generation.

2.4 Comparative Recommendation Explanations

In the view of a recommendation item, most of the previous approaches rely on evaluative explanations, assessing the quality of an individual item along some aspects of interest to the user. Less effort has been put into assessing a recommended item in comparison to another item or group of items. A traditional neighbor-based recommendation can produce explanation for an item similar to reference item but we have to judge how similar they are by comparing them ourselves. [5] studies a related form of comparative explanation, called tradeoff-oriented explanation, aiming at comparing product clusters, which is validated to be useful via a user study. [40] develops a comparative explainable recommendation incorporating comparative constraints from users' history of adoptions into explainable recommendation models. There are various problems related to comparisons that are not directly related recommendation yet can be considered for comparative explanations. One is to determine which of two products is better overall [42, 97]. For instance, it could be based on how two named entities are compared within the same sentence [78]. Another line is in finding substitute and/or complementary products. [57] relies on discovering topics in product reviews and networks of products derived from browsing and co-purchasing logs. Yet another related problem is competitor mining [28, 38, 90], finding which products are most likely to be comparable to a target product. In Chapter 5, we develop an explainable recommendation model that produces a template-based explanation that compare the recommendation product with another reference product that user purchased before. Chapter 6, we extend the comparison to multiple items by selecting comparative sets of reviews simultaneously.

¹<https://concept.research.microsoft.com/>

Part I

Evaluative Recommendation Explanations

Chapter 3

Synthesizing Explanation for Explainable Recommendation

In this chapter, we elaborate a *post-hoc* recommendation explanation approach for a set of compatible explainable recommendation models which employ aspect-level sentiments in their optimization objective, e.g., EFM [96] and MTER [85]. This framework aims at a more flexible, literate, and varied explanation covering various aspects of interest, rather than a standardized template, customized only via fill-in-the-blank aspect sentiments. We synthesize an explanation by selecting snippets (i.e., sentences) from reviews, while optimizing for representativeness and coherence. To fit target users' aspect preferences, we contextualize the opinions based on a compatible explainable recommendation model. Experiments on datasets of several product categories showcase the efficacies of our method as compared to baselines based on templates, review summarization, selection, and text generation.

3.1 Problem Formulation

Table 3.1 lists the notations used in this chapter. \mathcal{U} and \mathcal{P} are the universal sets of m users and n products respectively. User $u_i \in \mathcal{U}$ may assign to a product $p_j \in \mathcal{P}$ a rating $r_{ij} \in \mathbb{R}_+$ and

$\mathcal{U}, \mathcal{P}, \mathcal{A}, \mathcal{O}, \mathcal{T}$	set of all users, items, aspects, opinions, and reviews
$\mathcal{T}_{:j} \in \mathcal{T}$	set of observed text reviews on product p_j
$\mathcal{S}_{:j} \subseteq \mathcal{T}_{:j}$	set of all sentences on product p_j
$\mathcal{T}_{i:} \in \mathcal{T}$	set of observed text reviews of user u_i
$\mathcal{S}_{i:} \subseteq \mathcal{T}_{i:}$	set of all sentences of user u_i
$t_{ij} \in \mathcal{T}_{:j} \cap \mathcal{T}_{i:}$	a review of user u_i on product p_j
\mathcal{M}	explainable recommendation model
Z	aspect-level sentiments
$z_{ijk} \in Z$	sentiment of user u_i on item p_j about aspect a_k
\mathcal{D}	aspect demand
τ	solution set of selected sentences
$\Gamma_{ss'}$	variable indicates whether sentence s representing s'
γ_s	variable indicates whether sentence s is selected
$\zeta_{i'}$	variable indicates whether a review $t_{i'j}$ is part of τ
$\sigma_{si'}$	observed indicator of whether sentence s is in $t_{i'j}$
π_{sk}	observed indicator of whether sentence s expresses a_k
$s(w)$	sentence s after substituting opinion phrase w

Table 3.1: Main Notations

a text review t_{ij} . Let R be the observed user-item rating matrix, and \mathcal{T} be the set of observed text reviews. Let \mathcal{A} and \mathcal{O} be the universal sets of aspects and opinion phrases. We assume the occurrence of aspect $a \in \mathcal{A}$ and opinion phrase $o \in \mathcal{O}$ can be detected from a review sentence as described in [96].

Compatible Recommendation Models. Our objective is to synthesize an explanation based on the outputs of compatible explainable recommendation models (see Section 3.4 for examples). An explainable recommendation model \mathcal{M} produces both personalized recommendations and aspect-level sentiments $Z \in \mathbb{R}_+^{m \times n \times v}$ to facilitate their explanations. $z_{ijk} \in Z$ indicates user u_i 's sentiment for aspect a_k of p_j .

Problem Statement. Given aspect-level sentiments Z , and a product p_j recommended to user u_i by a model \mathcal{M} , we output an explanation in the form of a collection of sentences τ based on the aspect demand \mathcal{D} . Let aspect demand $\mathcal{D} \in \mathbb{N}^v$ be a vector, where each element \mathcal{D}_k is a non-negative integer indicating the number of sentences demanded for aspect $a_k \in \mathcal{A}$, and $v = |\mathcal{A}|$.

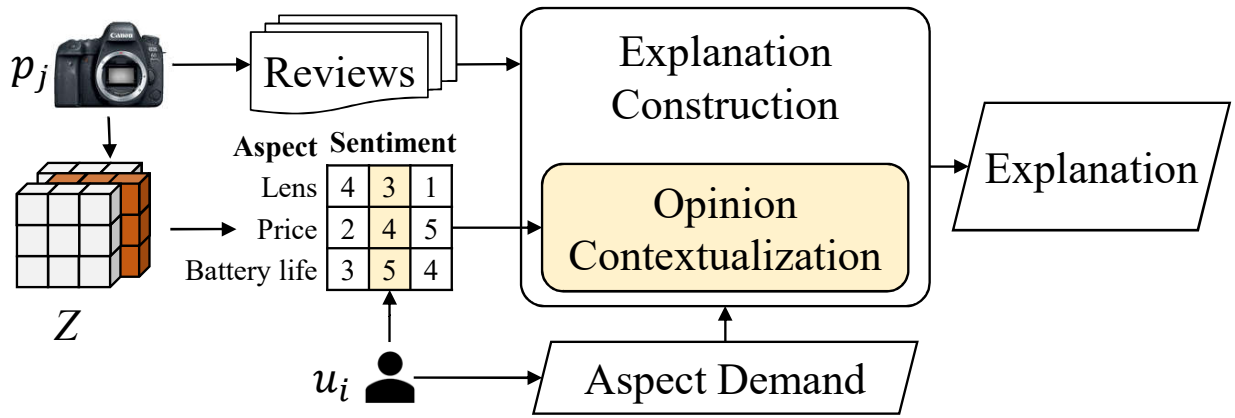


Figure 3.1: Architecture of proposed framework SEER

It follows that the sentences should reflect the aspect-level sentiments of the user specified in Z .

Evaluation. A question arises on how to evaluate a recommendation explanation, aside from the goal of meeting the aspect demand. In the literature, recommendation accuracy is measured in terms of how well the prediction approaches the ground truth (held-out rating). An analogous approach would then be to compare an explanation against a ground truth. Intuitively, the review that a user writes for a product *a posteriori* would have been a “perfect” explanation if we were recommending the same product *a priori*. Thus, in the experiments we will compare synthesized explanations in terms of similarity to held-out reviews.

3.2 SEER Framework

As seen in Figure 3.1, our framework is to synthesize an explanation by selecting snippets (sentences) from a product’s existing reviews. Here we discuss the objective of the selection, and offer optimal as well as approximate formulations.

3.2.1 Optimization Objective

When recommending product p_j to user u_i , we construct an explanation from $\mathcal{T}_{:j}$ (reviews of product p_j). The solution τ ideally consists of \mathcal{D}_k sentences for each $a_k \in \mathcal{A}$, selected from review sentences $\mathcal{S}_{:j}$ (the union of sentences from $\mathcal{T}_{:j}$).

Representativeness. To explain the aspect a_k of p_j well, we aim for the most representative among sentences in $\mathcal{S}_{:j}$ pertinent to a_k . Suppose that how well a sentence s could “represent” another sentence s' is reflected by a cost $\delta_{ss'} \in \mathbb{R}_+$ (lower is better). This may encode application-specific semantic notion of similarity, and for generality we consider these as a given. In Section 3.5, we experiment with several definitions, including unsupervised (e.g., cosine similarity between *tfidf* vectors), as well as supervised notions (e.g., paraphrase identification, textual entailment [34]).

Our task is to select \mathcal{D}_k most representative ones to place into the solution set τ . To encode this selection, let $\Gamma_{ss'}$ be a binary variable (the outcome to be determined) indicating whether a selected sentence $s \in \tau$ (i.e., $\gamma_s = 1$) represents another sentence $s' \in \mathcal{S}_{:j}$. We thus want to minimize the representation cost below, where we prefer a solution τ with sentences similar to many of the same aspect.

$$\text{r_cost}(\tau) = \sum_{s \in \tau} \sum_{s' \in \mathcal{S}_{:j}} \delta_{ss'} \cdot \Gamma_{ss'} \quad (3.1)$$

Coherence. In addition to capturing the aspects well, the explanation should be compact and coherent. Intuitively, a document by fewer authors would be more coherent than by many. Hence, we attach a cost $\theta_{i'}$ (given) to using a review $t_{i'j} \in \mathcal{T}_{:j}$, rather than to individual sentences. This way, the selection favors selecting sentences that may have come from the same review, presumably enhancing coherence. We define the coherence cost below, where $\zeta_{i'}$ is a binary variable of whether a review $t_{i'j} \in \mathcal{T}_{:j}$ (i.e., $\zeta_{i'} = 1$) is part of the solution set τ (i.e., one or more

of its sentences are selected).

$$\text{c_cost}(\tau) = \sum_{t_{i'j} \in \mathcal{T}_{:j}} \theta_{i'} \cdot \zeta_{i'} \quad (3.2)$$

The given cost $\theta_{i'}$ also serves to contextualize the explanation to a specific user, as defined shortly in Section 3.3.

Overall Cost. The overall cost is thus:

$$\text{cost}(\tau) = \text{c_cost}(\tau) + \text{r_cost}(\tau) \quad (3.3)$$

The two components have an inherent trade off. Adding a sentence may lower r_cost if the new sentence is more similar to other sentences, but that risks increasing the c_cost if the new sentence comes from a review not currently in the solution. On the other hand, fewer reviews may constrain the selection of representative sentences. Hence, we need an effective algorithm to find the optimal aggregate of the two.

3.2.2 Optimal Formulation via ILP

To find an optimal solution τ , we express the problem as Integer Linear Programming (ILP). (3.4a) is the objective (Eq. 3.3). γ_s is a binary indicator whether the sentence $s \in \mathcal{S}_{:j}$ is a part of τ . Constraints (3.4b) and (3.4c) ensure that sentence $s' \in \mathcal{S}_{:j}$ must be represented by one of the sentences s in the solution set ($\gamma_s = 1$). (3.4d) means a review must be selected when we select any of its sentences. $\sigma_{si'}$ is an observed binary indicator of whether s is in the review $t_{i'j}$. (3.4e) ensures a sentence is represented by another of the same aspect. Binary π_{sk} indicates whether s

is of aspect a_k . (3.4f) satisfies aspect demand.

$$\min: \sum_{t_{i'j} \in \mathcal{T}_j} \theta_{i'} \cdot \zeta_{i'} + \sum_{s, s' \in \mathcal{S}_{:j}} \delta_{ss'} \cdot \Gamma_{ss'} \quad (3.4a)$$

$$\text{s.t.} \sum_{s \in \mathcal{S}_{:j}} \Gamma_{ss'} = 1, \forall s' \in \mathcal{S}_{:j} \quad (3.4b)$$

$$\Gamma_{ss'} \leq \gamma_s, \forall s, s' \in \mathcal{S}_{:j} \quad (3.4c)$$

$$\gamma_s \cdot \sigma_{si'} \leq \zeta_{i'}, \forall t_{i'j} \in \mathcal{T}_{:j}, s \in \mathcal{S}_{:j} \quad (3.4d)$$

$$\Gamma_{ss'} \leq \sum_{a_k \in \mathcal{A}} \pi_{sk} \cdot \pi_{s'k}, \forall s, s' \in \mathcal{S}_{:j} \quad (3.4e)$$

$$\sum_{s \in \mathcal{S}_{:j}} \gamma_s \cdot \pi_{sk} = \mathcal{D}_k, \forall a_k \in \mathcal{A} \quad (3.4f)$$

$$\zeta_{i'}, \gamma_s, \Gamma_{ss'} \in \{0, 1\}, \forall t_{i'j} \in \mathcal{T}_{:j}; \forall s, s' \in \mathcal{S}_{:j} \quad (3.4g)$$

NP-hardness. Though SEER-ILP is theoretically optimal, it may be intractable for large problem sizes.

Proof. The proof sketch is based on a reduction from the Uncapacitated Facility Location Problem (UFLP) [13] involving a set of facilities and a set of customers. There is a cost to open each facility (favoring fewer facilities) and a cost to serve a customer from an open facility (favoring facility closer to customer). We reduce UFLP to our problem where there is only a single aspect. Each customer is now a sentence s' to be represented. Each facility is a review with opening cost $\theta_{i'}$, associated with one representing sentence s . The service cost is thus $\delta_{ss'}$. Our problem specifies the number of sentences to be selected for that aspect. If we solve for all demands from 1 to m , where m is the total number of facilities, we arrive at a solution for UFLP with the lowest total cost at any number of facilities. Since UFLP is known to be NP-hard, our more general formulation is NP-hard. \square

Algorithm 1 SEER-Greedy

```
1: Initialize  $\tau = \emptyset; S = \mathcal{S}_{:j}; T = \mathcal{T}_{:j}; D = \mathcal{D};$ 
2: while  $S \neq \emptyset$  do
3:   for  $t_{i'j} \in T$  do
4:     Find  $\tau_{i'} \subseteq t_{i'j}$  that represent the most number of unmet aspects in  $D$ , which minimize
       the average covering cost of sentences:  $\frac{\theta_{i'} + \sum_{s \in \tau_{i'}} \sum_{s' \in S} \delta_{ss'} \cdot \Gamma_{ss'}}{\sum_{s \in \tau_{i'}} \sum_{s' \in S} \Gamma_{ss'}}$ 
5:      $\tau := \tau \cup \tau_{i'}; T := T \setminus t_{i'j}$ 
6:      $S := S \setminus S'$ , where  $S'$  are  $\tau_{i'}$  covering sentences
7:      $D := D \setminus \{a\}$ , where  $\{a\}$  are  $\tau_{i'}$  representing aspects
8: return  $\tau$ 
```

3.2.3 Approximation via Greedy Algorithm

We therefore seek an approximation to cater to large problems. Non-metric UFLP has a greedy solution [25] with an approximation ratio of $1 + \log(n)$ based on a mapping to Minimum Weight Set Cover (MWSC). Our problem is different from UFLP in several respects, chiefly the aspect demands, precluding direct reuse of that particular greedy solution. Even when confined to one aspect, there is no existing solution with provable guarantee for MWSC with constraint on the number of sets [20].

Our proposed greedy solution is Algorithm 1. Sentences in $\mathcal{S}_{:j}$ are the coverable elements. A covering set is a review $t_{i'j}$ with its selected sentences $\tau_{i'}$ to cover a subset of S ; its weight is

$$\frac{\theta_{i'} + \sum_{s \in \tau_{i'}} \sum_{s' \in S} \delta_{ss'} \cdot \Gamma_{ss'}}{\sum_{s \in \tau_{i'}} \sum_{s' \in S} \Gamma_{ss'}}$$

Enumerating all subsets is exponential. In practice, it is sufficient to sort $s' \in S$ in terms of $\delta_{ss'}$ and investigate the first k sentences for various k [25]. We greedily pick the lowest-weight set until all the sentences are covered.

Unique to our scenario is the selection of $\tau_{i'}$ from the sentences in $t_{i'j}$, by maximizing the representation of aspects, which always lowers the cost of representation. If there are multiple sentences that can represent an aspect, we seek the permutation with the lowest cost. To ensure coverage, the last sentence should cover all remaining sentences of the aspect.

Complexity Analysis. In Algorithm 1, the two outer loops (lines 2–3) may require $O(|\mathcal{S}_{:j}| \cdot |\mathcal{T}_{:j}|)$. The inner cost is dominated by line 4. Computing the cost is $O(t_{avg} \cdot |\mathcal{S}_{:j}|)$, where t_{avg} is the average length of reviews. Sorting the covered sentences is $O(|\mathcal{S}_{:j}| \log |\mathcal{S}_{:j}|)$. Since $t_{avg} \cdot |\mathcal{T}_{:j}|$ is equivalent to $|\mathcal{S}_{:j}|$, the overall complexity of SEER-Greedy is $O(|\mathcal{S}_{:j}|^3 + |\mathcal{T}_{:j}| |\mathcal{S}_{:j}|^2 \log |\mathcal{S}_{:j}|)$.

3.3 Opinion Contextualization

The goal is an explanation with compatible opinions to the ones the target user would have (as encoded in the Z). Contextualizing the sentences to fit the target user’s aspect sentiments is done via two complementary mechanisms.

Sentence Selection. One means is to employ $\theta_{i'}$ that favors more compatible reviewers in Equation 3.2. $\theta_{i'}$ is defined as a function of the similarity between z_{ij} : (a vector of aspect-level sentiments by target user u_i on p_j) and $z_{i'j}$., e.g.,

$$\theta_{i'} = \frac{1 - \cos(z_{ij}, z_{i'j})}{2}$$

Alternatively, our framework could admit other definitions for $\theta_{i'}$ as well.

Opinion Substitution. To “extend” beyond the original pool of review sentences, we contextualize candidate sentences by allowing substitution of the original opinion phrase with another more attuned to the target user’s sentiments. After removing the opinion to be substituted, this turns into a sentence completion task, which is an NLP problem in its own right. For concreteness, we allude to a specific solution, but a fuller consideration is beyond the scope of this work. *Context2Vec* [58] pays attention to the entire sentential context, with two LSTMs for sentence-level representation: one reads from the left (ILS) and the other from the right (rLS). Their concatenation passes through a 2-layer perceptron with ReLU activation to get its context

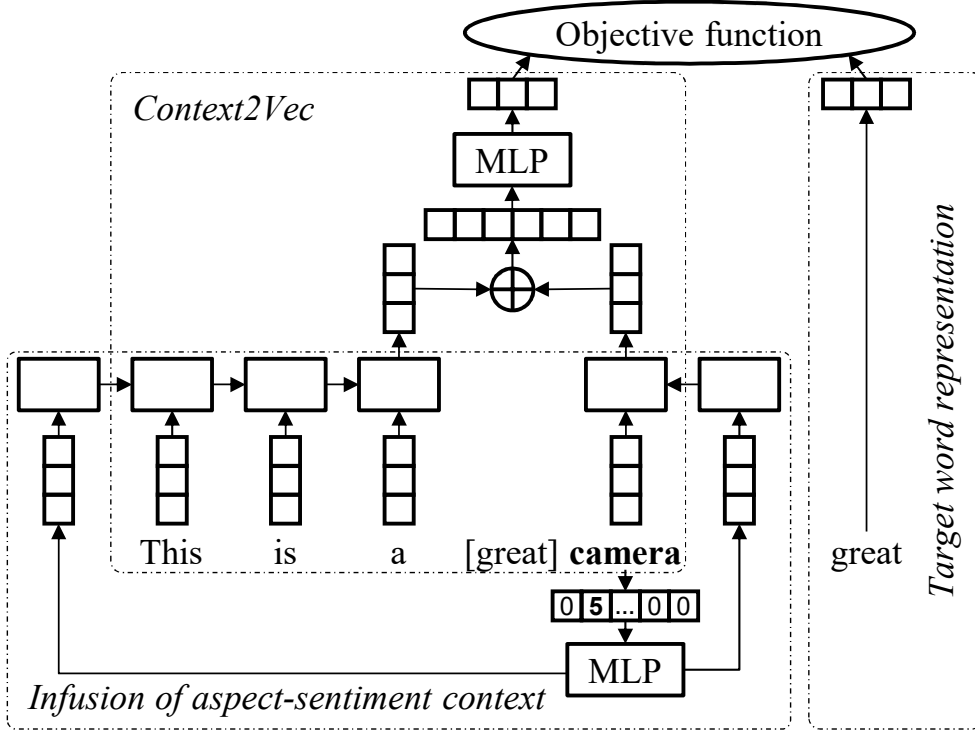


Figure 3.2: ASC2V Architecture

representation. L_1, L_2 are fully connected linear operations.

$$\vec{w}_l = L_2(\text{ReLU}(L_1(\text{ILS}(w_{1:l-1}) \oplus \text{rLS}(w_{|s|:l+1}))))$$

As *Context2Vec* only considers the surrounding words, the sentence completion is irrespective of the user’s aspect-level sentiment. To “personalize” the explanation, we use our modification, called *Aspect-Sentiment Context2Vec* or *ASC2V*, for predicting opinionated word based on sentence context, and z_{ijk} , i.e., u_i ’s sentiment for aspect a_k of p_j . To infuse this information explicitly, we construct an aspect-sentiment vector $\vec{a}s$ of dimensionality $|\mathcal{A}|$. If the sentence is of aspect a_k , we set the k^{th} dimension to the value of z_{ijk} , and 0 otherwise. We use 1-layer perceptron with tanh activation to project aspect sentiment information into the same space as context word embedding. L_3 is fully connected linear operation.

$$\vec{w} = \text{tanh}(L_3(\vec{a}s))$$

Algorithm 2 Opinion Substitution

```
1: Initialize  $min_{r\_cost} := r\_cost(\tau)$ 
2: for  $s \in \tau$  do
3:    $\tau' := \tau \setminus \{s\}$ ;  $w_{best} := get\_opinion(s)$ 
4:   for  $w \in O_{ijs}$  do
5:      $current\_cost := r\_cost(\tau' \cup \{s(w)\})$ 
6:     if  $current\_cost < min_{r\_cost}$  then
7:        $w_{best} := w$ ;  $min_{r\_cost} := current\_cost$ 
8:    $\tau := (\tau \setminus \{s\}) \cup \{s(w_{best})\}$ 
9: return  $\tau$ 
```

This \vec{w} is the starting token for both ILS and rLS (see Figure 3.2). We rank candidate opinions based on cosine similarity of their embeddings with the context vector. For the example “This is a ___ camera”, if z_{ijk} expresses positive sentiment, “great” should be ranked highly. If negative, a different opinion may apply.

ASC2V contextualizes sentences within the synthesized explanation to further improve the objective in Equation 3.3. Let O_{ijs} be top- k predicted opinions for sentence s based on *ASC2V* (for experiments, we use $k = 10$). As shown in Algorithm 2, we substitute each opinion phrase $w \in O_{ijs}$ (line 4) into s by $s(w)$ and keep the one minimizing r_cost (line 7). c_cost is not affected as only the opinion, but not the sentence, changes. This computation is efficient at $O(|\tau| \cdot k)$, as the solution size $|\tau|$ and number of opinions k are usually relatively small.

3.4 Compatible Recommendation Models

The class of compatible models are broadly defined. [3, 96] are based on matrix factorization, while [6, 85] are based on tensor factorization. Others combine matrix factorization with topic modeling [88]. Several works enhance their explainable models by using graphs [23] or trees [18]. As concrete examples, in Section 3.5, we experiment with two models, EFM and MTER, which were established methods for templated explanations.

Explicit Factor Model or *EFM* [96] reconstructs the observed rating matrix R , user attention matrix X , and product quality matrix Y . Each $x_{ik} \in X$ indicates the importance of aspect a_k to user u_i , while each $y_{jk} \in Y$ is the summative quality of product p_j on aspect a_k . EFM decomposes the observations X, Y , and R into latent factors, minimizing the function

$$\|PQ^T - R\|_F^2 + \lambda_x \|\eta_1 \psi^T - X\|_F^2 + \lambda_y \|\eta_2 \psi^T - Y\|_F^2$$

where $P = [\eta_1 \ \phi_1]$ and $Q = [\eta_2 \ \phi_2]$ are users' and products' latent factors respectively. Each is the concatenation of aspect-based factors (η_1, η_2) influenced by X, Y and hidden factors (ϕ_1, ϕ_2) influenced by ratings. ψ are the latent factors of aspects. Coefficients λ_x and λ_y weigh the relative importance of aspects vs. ratings. We derive Z from the Hadamard product of the reconstructions \hat{X}, \hat{Y} , i.e., $z_{ijk} = \hat{x}_{ik} \times \hat{y}_{jk}$.

Multi-Task Explainable Recommendation or *MTER* [85] models user-product-aspect interactions jointly as a tensor G , where $g_{ijk} \in G$ reflects the aggregate sentiment scores across all mentions by user u_i of aspect a_k in product p_j 's reviews. The rating r_{ij} is appended as an additional aspect to the tensor G , i.e., $g_{ijv} = r_{ij}$. G is decomposed using Tucker decomposition [31]. Let \hat{G} be its reconstruction after minimizing the function

$$\|\hat{G} - G\|_F - \lambda \sum_{u_i \in \mathcal{U}} \sum_{(u_i, p_j, p'_j)} \ln \sigma(\hat{g}_{ijv} - \hat{g}_{ij'v})$$

where (u_i, p_j, p'_j) is a pairwise ranking observation where u_i prefers p_j to p'_j . We synthesize an explanation based on the non-rating aspects of \hat{G} , i.e., $z_{ij(0:v-1)} = \hat{g}_{ij(0:v-1)}$.

3.5 Experiments

Comparisons are tested with one-tailed paired-sample Student's t-test at 0.05 level. Experiments were run on machine with Intel Xeon E5-2650v4 2.20 GHz CPU and 256GB RAM.

Dataset	#User	#Product	#Aspect	#Opinion	#Review	#Sentence	$\frac{\#Review}{\#Product}$	$\frac{\#Sentence}{\#Review}$
Computer	19,818	8,606	5,354	4,243	163,894	512,703	19.04	3.13
Camera	4,770	2,680	2,321	2,367	37,856	151,382	14.13	3.99
Toy	2,672	1,984	818	1,225	26,598	57,260	13.41	2.15
Cellphone	2,340	1,390	882	1,256	19,109	51,469	13.75	2.69

Table 3.2: Data statistics

Dataset	EFM				MTER			
	Coverage	Overall Cost	Solve Time	# Optimal Solution	Coverage	Overall Cost	Solve Time	# Optimal Solution
Computer	100.00	100.83	4.37	95.07	100.00	100.85	4.05	95.11
Camera	100.00	100.98	4.07	95.55	100.00	100.78	3.67	95.64
Toy	100.00	100.62	2.86	99.95	100.00	100.11	2.19	99.95
Cellphone	100.00	100.81	3.79	98.05	100.00	100.31	3.12	98.05
Total	100.00	100.84	4.16	95.73	100.00	100.74	3.78	95.77

Table 3.3: Performance ratios of SEER-Greedy to SEER-ILP (%)

Datasets. Experiments use four public datasets of Amazon reviews¹ [56] of varying categories: *Computer and Accessories* (Computer), *Camera and Photo* (Camera), *Toys and Games* (Toy), *Cell Phones and Accessories* (Cellphone). For each category, we filter out users and items with fewer than five reviews. The remaining are split into training, validation, and test at a ratio of 0.6 : 0.2 : 0.2 for every user chronologically. Sentences in validation and test with opinions or aspects that had not appeared in training were excluded. Table 3.2 shows some basic statistics of the datasets.

Base Models. SEER uses aspect-level sentiments Z from two compatible explainable recommendation models. For EFM², as in the original work, the latent factor and explicit factor dimensions are 60 and 40. For MTER, we adopt the default setting of the author’s implementation³. It is not our intention to compare these two, as our model works with any compatible base recommendation method.

¹<http://jmcauley.ucsd.edu/data/amazon/>

²<https://github.com/PreferredAI/cornac>

³<https://github.com/MyTHWN/MTER>

		Computer	Camera	Toy	Cellphone
EFM	SEER _{tfidf}	15.14 [§]	14.74 [§]	16.36 [§]	14.96 [§]
	SEER _{SSE}	14.48	14.01	15.39	14.40
	SEER _{ESIM}	13.80	13.51	14.81	14.10
MTER	SEER _{tfidf}	15.15 [§]	14.71 [§]	16.28 [§]	15.03 [§]
	SEER _{SSE}	14.49	14.03	15.37	14.42
	SEER _{ESIM}	13.79	13.52	14.84	14.10

[§] denotes statistically significant improvements. Highest values are in **bold**

Table 3.4: Comparison of representative costs: ROUGE-L

Evaluation Metrics. We use ROUGE [46], a well-known metric for text matching and text summarization, to assess how well the synthesized explanations approach the ground-truth reviews. To cater to words as well as phrases, we report ROUGE-1 (1-gram) as well as ROUGE-L (longest common subsequence) summatively in terms of the F-Measure.

3.5.1 Explanation Synthesis

Optimal vs. Approximation. For the optimal SEER-ILP, within 100 seconds, the CPLEX⁴ solver can solve optimally for $\geq 95\%$ of problem instances. Running on the same instances, SEER-Greedy achieves identical coverage of aspects (100%) at an overall cost that is just 1% higher than optimal, yet consumes merely 4% (i.e., a couple of seconds) of the time taken by SEER-ILP on average (see Table 3.3). Subsequently, we run both variants on 100% of the problem instances. For ILP, the result would reflect either the optimal or the best solution up to that point.

Representativeness Cost. For the representativeness cost $\delta_{ss'}$ in Equation 3.1, we explore several options. One is based on the cosine similarity of sentences s and s' . Each sentence is represented by *tfidf* vectors based on the vocabulary of product’s sentences. For $\delta_{ss'}$, we take

$$\delta_{ss'} = \frac{1 - \cos(s, s')}{2}$$

⁴<https://www.ibm.com/analytics/cplex-optimizer>

Model	Computer		Camera		Toy		Cellphone	
	MRR	R@10	MRR	R@10	MRR	R@10	MRR	R@10
C2V	0.460	0.695	0.411	0.645	0.515	0.705	0.365	0.621
RC2V	0.462	0.706	0.409	0.643	0.514	0.707	0.366	0.624
ASC2V _{EFM}	0.475 [§]	0.713 [§]	0.416 [§]	0.652 [§]	0.526 [§]	0.726 [§]	0.384 [§]	0.649 [§]
ASC2V _{MTER}	0.473 [§]	0.709 [§]	0.418 [§]	0.653 [§]	0.528 [§]	0.724 [§]	0.388 [§]	0.651 [§]

[§] denotes statistically significant improvements by ASC2V

Highest values (among ASC2V, RC2V, and C2V) are in **bold**

Table 3.5: Opinion Contextualization

We also try two other models: SSE [63] for paraphrase identification and ESIM [7] for textual entailment. Table 3.4 shows *tfidf* to perform the best in terms of ROUGE-L. We will use it subsequently. One reason is the corpus SSE and ESIM trained on was not optimized for review sentences. In any case, we consider $\delta_{ss'}$ as given.

3.5.2 Opinion Contextualization

We hide the ground-truth opinion from the held-out test review and evaluate the ranking of candidates in \mathcal{O} using IR metrics: MRR (the reciprocal rank of the true opinion, averaged across held-out reviews) and Recall@10 or R@10 (fraction of held-out reviews with the true opinion in the top-10).

We compare our ASC2V with two baselines. *Context2Vec* or C2V [58] with only on the sentence (no aspect sentiment). RC2V uses random aspect sentiment. For ASC2V, we train with similar setting as C2V, using RMSprop for optimization. Table 3.5 shows both variants of ASC2V significantly outperform C2V. RC2V, which adds no meaningful information, fluctuates around C2V. Indeed aspect-level sentiments are useful for opinion contextualization.

3.5.3 Comparison to Baselines

We compare the explanations generated by SEER to several categories of baselines. For parity, we control for the explanation length. The first category is *template explanation*, comprising the

Model		Computer	Camera	Toy	Cellphone
ATT2SEQ		0.192	0.162	0.257	0.195
EXPANSION NET		0.478	0.612	0.734	0.504
AP-REF2SEQ		0.212	0.242	0.367	0.242
TEXT RANK		0.234	0.219	0.311	0.266
REPRESENTATIVE		0.408	0.407	0.480	0.448
COMPREHENSIVE		0.678	0.629	0.717	0.678
CHARACTERISTIC		0.153	0.169	0.291	0.207
CHARACTERISTIC+		0.574	0.521	0.662	0.582
EFM	TEMPLATE	0.697	0.654	0.725	0.687
	SEER-Greedy	0.775 [§]	0.729 [§]	0.787 [§]	0.768 [§]
	SEER-ILP	0.775 [§]	0.729 [§]	0.787 [§]	0.768 [§]
MTER	TEMPLATE	0.775 [§]	0.729 [§]	0.787 [§]	0.768 [§]
	SEER-Greedy	0.775 [§]	0.729 [§]	0.787 [§]	0.768 [§]
	SEER-ILP	0.775 [§]	0.729 [§]	0.787 [§]	0.768 [§]

[§] denotes statistically significant improvements. Highest values are in **bold**

Table 3.6: Comparison to Baselines: Coverage

original explanations by EFM [96] and MTER [85]. Next is *review summarization* represented by TEXT RANK [2] and *review selection* with four methods: REPRESENTATIVE selects the review with lowest representative cost (see Equation 3.1); COMPREHENSIVE selects the review of highest aspect coverage [82]; CHARACTERISTIC selects the review whose aspect sentiment distribution most resembles a product’s reviews [37]; CHARACTERISTIC+ that also takes into account the aspect demand by considering distributions of demanded aspects only. The last category is *review generation* with ATT2SEQ [16] that generates text from user, product, and rating as attributes; EXPANSION NET [61] that generates text from aspect words as starter phrases; and AP-REF2SEQ [62] that generates text from user & item reviews and aspect words.

Coverage. Table 3.6 shows the coverage, i.e., the proportion of the met aspect demand. Coverage is not necessarily 1 due to the limited number of candidate sentences for selection or aspects that have not appeared before. Both SEER variants outperform baselines in coverage (MTER has identical coverage). The template methods respond to aspect demand. EFM produces duplicate sentences for an aspect, resulting in lower coverage than MTER that produces multiple sentences by varying opinion phrases. Methods that do not benefit from aspect demands (in-

Model	Computer		Camera		Toy		Cellphone		
	R-1	R-L	R-1	R-L	R-1	R-L	R-1	R-L	
ATT2SEQ	16.69	10.35	15.90	9.13	16.51	10.41	16.42	9.76	
EXPANSION NET	11.68	1.25	19.23	5.19	24.41	4.68	14.13	3.11	
AP-REF2SEQ	16.94	12.29	17.04	12.94	21.72	14.50	19.15	12.99	
TEXT RANK	18.68	11.15	19.29	11.37	19.04	11.97	19.16	11.25	
REPRESENTATIVE	18.22	11.11	19.24	11.27	19.72	12.60	19.45	11.80	
COMPREHENSIVE	21.90	13.44	22.16	13.16	23.41	15.12	22.33	13.73	
CHARACTERISTIC	13.18	7.65	14.05	7.92	15.76	9.80	14.27	8.41	
CHARACTERISTIC+	18.33	10.87	18.06	10.32	21.25	13.56	19.05	11.34	
EFM	TEMPLATE	14.17	8.41	14.43	8.39	13.37	8.06	14.22	8.41
	SEER-Greedy	24.89 [§]	15.05 [§]	25.11 [§]	14.72 [§]	25.33 [§]	16.30 [§]	24.66 [§]	14.87 [§]
	SEER-ILP	25.12[§]	15.14 [§]	25.23[§]	14.74[§]	25.43[§]	16.36[§]	24.76[§]	14.96 [§]
MTER	TEMPLATE	16.88	11.68	16.43	11.14	13.22	12.03	17.61	11.86
	SEER-Greedy	24.90 [§]	15.08 [§]	25.01 [§]	14.65 [§]	25.25 [§]	16.24 [§]	24.74 [§]	14.94 [§]
	SEER-ILP	25.12[§]	15.15[§]	25.22 [§]	14.71 [§]	25.33 [§]	16.28 [§]	24.85 [§]	15.03[§]

[§] denotes statistically significant improvements by our models. Highest values in **bold**

Table 3.7: Comparison to Baselines: ROUGE-1 and ROUGE-L

cluding ATT2SEQ, TEXT RANK, REPRESENTATIVE, and CHARACTERISTIC) underperform the other methods that do. Review selection methods are limited to what can be covered by a review. Among these, COMPREHENSIVE achieves the highest aspect coverage. As the review with the closest aspect sentiment distribution does not necessarily have the highest aspect coverage, the coverage of CHARACTERISTIC+ is lower than COMPREHENSIVE.

Ground Truth Recovery. As Table 3.7 shows, SEER variants (ILP and Greedy) significantly outperform all the baselines, with the highest F-Measure for both ROUGE-1 (R-1) and ROUGE-L (R-L)⁵. The template-based approaches perform poorly because a standard template cannot reflect varied reviews. Benefitting from paying attention to the aspect demand, CHARACTERISTIC+ performs better than CHARACTERISTIC. However, both still perform worse than COMPREHENSIVE that maximizes coverage of aspect demand. REPRESENTATIVE outperforms CHARACTERISTIC since it optimizes for representativeness yet is still lower than COMPREHENSIVE.

⁵We have experimented with other ROUGE variations (ROUGE-[1,2,L],S[1-4],SU[1-4]) with consistent results. SEER outperforms other baselines significantly in term of F-Measure. For conciseness, we report only the F-measure of ROUGE-1 and ROUGE-L.

User	A3ALXLASGICTBU	
Product	B002DPUUKK	
Title	Microsoft Wireless Mobile Mouse 4000 - White	
Ground truth	The mouse has worked great for about 1-year. The mouse was great for a while. The size is perfect for my hand	
ATT2SEQ	I really like the mouse. The mouse is very comfortable and the mouse is fine. I haven't had any problems with the wireless signal	
EXPANSION NET	The mouse is a plus. The size is great and the size is perfect	
AP-REF2SEQ	It's a good wireless mouse for the price. It's a good wireless mouse. It's a good mouse for the price	
TEXT RANK	This is a great mouse. A great mouse. Very good mouse	
REPRESENTATIVE	If you call up with a problem mouse that requires a replacement. An all black mouse is difficult to find inside a laptop bag in the dark. I selected the "downtown" version with the white glossy center panel and "city grid/skyline" motif	
COMPREHENSIVE	A great mouse. 4 stars instead of 5 because of the lightness and the smooth mouse wheel instead of the ratcheting one. The size is good	
CHARACTERISTIC	I got this mouse instead of a 3000 series because of the extra button on the side. The side button is not handy because of how it is placed so high and forward on the mouse. In which case you might not mind	
CHARACTERISTIC+	Thinking a wireless mouse would be good	
EFM	TEMPLATE	You might be interested in [mouse], on which this product performs well. You might be interested in [size], on which this product performs well
	SEER-ILP	The <mouse> is very [comfortable] and nice looking. This is a [great] <mouse>. The <size> is [perfect]
MTER	TEMPLATE	Its <mouse> is [easy-to-adjust]. Its <mouse> is [lefty]. Its <size> is [awkward]
	SEER-ILP	The <mouse> is very [comfortable] and nice looking. This is a [great] <mouse>. The <size> is [good]

Table 3.8: Example Explanations on a Computer instance

	Model	Annotator					Average
		1	2	3	4	5	
Q1	MTER	2.10	2.35	2.75	3.00	2.35	2.51
	AP-REF2SEQ	3.15	3.00	3.60	3.75	3.50	3.40
	SEER-ILP	3.95[§]	4.25[§]	4.00[§]	3.85[§]	4.10[§]	4.03[§]
Q2	MTER	1.75	1.50	2.40	2.80	2.00	2.09
	AP-REF2SEQ	1.95	3.05	3.20	3.40	3.10	2.94
	SEER-ILP	3.55[§]	4.45[§]	3.80[§]	3.75[§]	4.45[§]	4.00[§]

[§] denotes statistically significant improvements. Highest values are in **bold**

Table 3.9: Result analysis of user study

TEXT RANK underperforms COMPREHENSIVE, because of redundant sentences that repeat aspects while the latter considers a whole review covering various aspects. The review generation approaches tend to produce short and repetitive sentences. They do not reflect aspect-level sentiments fully: ATT2SEQ uses ratings but no aspects, whereas EXPANSION NET and AP-REF2SEQ use aspects but may not reflect sentiments well.

3.5.4 Qualitative Study

Case Study. As an illustration, Table 3.8 shows the explanations for a Computer instance. The ground truth review reveals aspect demand involving *mouse* and *size*. EFM describes the product having good performance on the two aspects. MTER opinions are difficult to understand. ATT2SEQ does not cover the aspect demand. EXPANSION NET generates short sentences repetitively. TEXT RANK tends to select popular repetitive aspects. Our SEER-ILP produces readable explanations that reflect not only the aspects, but also the user opinions. When used with EFM or MTER, it generates slightly different phrases, e.g., “perfect” vs. “good” *size*.

User Study. To test the efficacy of the explanations from human perspective, we randomly select 5 user-product pairs from each category to get 20 examples in total and design a survey involving five participants who are not the authors. There are two questions. The first looks into the language quality, e.g., readable and easy to understand. The second, which also appeared in

[85], looks into appropriateness of recommendation.

Q1: Are the explanatory sentences well-formed and understandable?

Q2: Does the explanation help you understand why the given product is being recommended to the given user?

Each question is applied to a given explanation. Each participant chooses from five-point Likert scale, from 1 (strongly disagree) to 5 (strongly agree). To compare to the proposed SEER-ILP, we choose MTER and AP-REF2SEQ as representative baselines, as these two were designed specifically for recommendation explanation and achieve high performance in terms of ROUGE-L. As reported in Table 3.9, SEER-ILP outperforms the two baselines significantly. For Q1, MTER with simple template is difficult to understand, while AP-REF2SEQ achieves better results (≥ 3) comparing to MTER which shows its ability to generate readable text. However, AP-REF2SEQ-generated text is short and too general which make their explanations less informative than those of SEER-ILP.

3.6 Summary

In this chapter, we propose an innovative post-hoc strategy for providing natural language explanations for personalized recommendations. Our approach synthesizes an explanation by selecting representative sentences from a product’s reviews, contextualizing the opinions based on aspect-level sentiments from a class of compatible explainable recommendation models. SEER performs well against competitive baselines including templates, review summarization, selection, and generation.

Chapter 4

Question-Attentive Review-Level

Recommendation Explanation

A ubiquitous feature of Web applications and e-commerce marketplaces today is a recommender system that aids users in navigating the multitude of options available, be they products to purchase, books to read, social media posts to view, movies to watch, etc. The most common framework is that of collaborative filtering [33], predicting ratings or adoptions based on users' past interactions with various items.

Earlier in the evolution of recommender systems, the concern was predominantly on achieving higher accuracies [24, 66]. Of late, the concern shifts to greater interpretability and explainability, as ultimately the goal is to get users to adopt the recommendations. This gives rise to a plethora of explainable recommendation models [94], which seek to produce not only recommendations, but also accompanying explanations. There are diverse forms of explanations, leveraging different types of information associated with either users or items.

For a pertinent instance, we allude to *review-level explanation*, whereby the explanation to a recommendation takes the form of a review, selected from the existing reviews of the product. An insightful review, when presented with a recommended product, allows the recipient of the recommendation to empathize with the hands-on experience of the reviewer, thus anticipating

Asin: B07P15K8Q7
Title: Canon EOS Rebel T7 DSLR Camera Bundle with Canon EF-S 18-55mm f/3.5-5.6 IS II Lens + 2pc SanDisk 32GB Memory Cards + Accessory Kit

Question: does this camera have wifi ability?>
Answer: Ya
 By Melaku D on May 14, 2019

Question: Why do you sell a flash with this bundle that does not work with the camera?
Answer: It works fine, the flash is used on the bar mount that is also provided. The bar mounts to the tripod thread on the bottom of the Camera and then the Flash mounts to the bar. It is for a secondary flash, hence it senses your popup flash and goes off. Technically you can have the flash behind the subject that way the ba... see more
 By Ultimaron on November 24, 2019

Question: Can this camera send photos to smartphones using WiFi?
Answer: Yes. You have to set up the Wi-Fi functions, nickname your camera. I really recommend the Canon EOS Rebel T7/2000D book (for dummies) book as a companion translation to the manual.
 By Wily Girl on June 26, 2019

Question: does it have a port for an external microphone?
Answer: No, only the t7i
 By Lane Wallen on January 20, 2021
 ~ See more answers (1)

Question voting See more answered questions (216)

★★★★★ **Good Starter kit**
 Reviewed in the United States on April 13, 2019
 Verified Purchase | Early Reviewer Rewards (What's this?)
 I purchased this as a gift but I own several Canon products. This is a good starter kit as it provides enough for the person to experiment and learn about photography and the equipment before deciding what els they may need. The person was pleasantly surprised by the amount of things that actually came in the package.

185 people found this helpful Review helpful voting

Helpful Report abuse

Question

Review

Figure 4.1: A product with question and review

what her own experience with the product would be. For instance, on Amazon.com, Canon EOS Rebel T7 Bundle¹ has more than 2800 ratings, more than 300 of which have reviews. One of these reviews is illustrated in Figure 4.1, relating to the quality of the starter kit. That popular products may have many reviews (some to the tune of tens of thousands) is a dual-edged sword. With a rich corpus for selection comes the problem in how to select which review to present as an explanation. One existing paradigm [8, 51] is to weigh the contribution of various reviews to the recommendation objective.

Given the abundance of reviews, there is a proclivity to employ reviews to aid recommendations. Most of the works are intent on improving recommendation accuracy rather than to

¹<https://www.amazon.com/Canon-T7-18-55mm-3-5-5-6-Accessory/dp/B07P15K8Q7/>

serve directly as explanations. These include content-based methods based on topic models [73], sentiments [14], social networks [67]. By using convolutional neural network, [99] encodes all reviews on an item to represent that item and all reviews written by a user to represent that user to enhance rating prediction. [76] learns to focus on a few reviews of users and items optimizing for rating prediction. In contrast to works that see reviews as content to help recommendation accuracy, we focus on the role of reviews as explanations.

In this work, we propose to go beyond reviews and incorporate other information associated with a product. One that is a focus of this work is a question posted by a user that in turn attracts answers from other users, hereinafter referred to in short form as QA. For instance, the same product Canon EOS Rebel T7 bundle featured in Figure 4.1 has more than 200 questions. Among them are whether the camera has wifi ability (*answer: yes*), whether there is a port for an external microphone (*answer: no, but another model T7i does*), and whether it is suitable for indoor sports (*answer: yes, it has a sport mode*). Similarly to reviews, QAs could also receive votes from users.

Interestingly, questions and their answers present a distinct yet complementary information to reviews. Where reviews tend to be subjective and replete with opinions, questions tend to be objective and inquisitive of factual concerns. Where a single review tends to be multi-faceted and comprehensive, each question tends to be concise and narrowly focused on a single aspect. Given this complementarity, we postulate that both QA and review could collectively serve as recommendation explanations. The former notifies the recommendee of relevant factual concern(s), while the latter gains the recommendee insights from a reviewer’s experience.

QA as a feature is also increasingly prevalent across many platforms, with Amazon.com and Tripadvisor.com being a couple of prominent examples. For instance, across the ten product categories in our datasets (see Section 4.2), between 13% to 56% of products have QA information. Given the anticipated further increase in QA data over time, it is timely to consider how to leverage QA in addition to reviews for more informative recommendation explanations.

\mathcal{U}, \mathcal{P}	set of all users and products
\mathcal{T}, \mathcal{Q}	set of all reviews and questions
$t_{ij} \in \mathcal{T}$	a review of user i on product j
Q_j	a set of all questions on product j
$q_{jk} \in Q_j$	a question k of product j
$\xi(t_{ij}), \xi(q_{jk})$	embedded matrices of t_{ij} and q_{jk}
$\zeta_u(i), \zeta_p(j)$	latent features of user i and product j
$O_{t_{ij}}, O_{q_{jk}}$	feature vectors extracted from t_{ij} and q_{jk}
u_i, p_j	rating-based representation of user i and product j
α_{ij}	attention weights for $O_{t_{ij}}$
β_{ijk}	attention weight of review t_{ij} on question q_{jk}
d_{jk}	document representation respecting to q_{jk}
γ_{jk}	attention weight of document d_{jk}
b_u, b_i, μ	user bias, item bias, and global bias respectively

Table 4.1: Main Notations

Problem. Let \mathcal{U} be a set of users, and \mathcal{P} be a set of products. A user $i \in \mathcal{U}$ assigns to a product $j \in \mathcal{P}$ a rating $r_{ij} \in \mathbb{R}_+$ along with a review t_{ij} . We denote the collection of all ratings as \mathcal{R} , that of all reviews as \mathcal{T} , and the subset of reviews concerning a product j as \mathcal{T}_j . Product j may also have multiple questions $\mathcal{Q}_j = \{q_{j1}, q_{j2}, \dots, q_{j|\mathcal{Q}_j|}\} \subset \mathcal{Q}$. Each question is presumed to be accompanied by answer(s), collectively referred to in short form as QA. Table 4.1 lists these notations and others to be introduced later.

The problem can thus be stated as follows. Receiving as input users \mathcal{U} , products \mathcal{P} , ratings \mathcal{R} , reviews \mathcal{T} , and question-answer pairs \mathcal{Q} , we seek a model capable of predicting a missing rating by a user i on product j for recommendation (rating regression), as well as identifying a question-answer pair (selected from \mathcal{Q}_j) along with a review (selected from \mathcal{T}_j) to serve collectively as explanations accompanying the recommendation.

Due to the differing yet complementary natures of QA and reviews, we design a neural attention model, called QUESTER, that operates at two levels. First, the concise QA serves as focal points of attention representing salient aspects to a product recommendation. Second, the multifaceted nature of reviews means that they could be relevant to multiple aspects, and we model their relative importance to each QA. Together, QA and reviews serve dual roles in a hand-in-

hand manner: to contribute content features to aid recommendation and to serve as explanations to a recommendation.

The use of QA for recommendation is still relatively rare in the literature. One is to detect a user’s propensity to purchase a product based on the question that the user has submitted [11]. This is a distinct scenario from ours where the question does not have to be posed by the recipient of recommendations. Rather, we see questions as additional product information that may be relevant as explanation. QA-based recommendation is also orthogonal from question answering task. [98] selects relevant sentences in product reviews to answer a question. [91] incorporates aspect on reviews for predicting answer of a yes-no question. Our goal is not to answer questions, rather to select QA appropriate for recommendation explanations.

Contribution. We make several contributions. *First*, to our best knowledge, this is the first work to incorporate product questions into an attention mechanism on reviews for recommendation. *Second*, we develop a neural model called QUESTion-attentive review-level Explanation for neural rating Regression or QUESTER, which considers questions as a source of alignment to textual review. An important question would help to identify important reviews. *Third*, we conduct comprehensive experiments on ten product categories against comparable baselines. Importantly, we find that not only do QAs help in identifying useful reviews, but the expanded explanation that is the combination of QA and review also has value.

4.1 Methodology

Our formulation in having a pair of QA and review to accompany recommendation based on rating regression is novel. We hypothesize that the concise questions could serve as an attention mechanism in weighing the importance of reviews. This achieves an alignment between questions and reviews, potentially allowing expanded explanations that are more comprehensive and coherent.

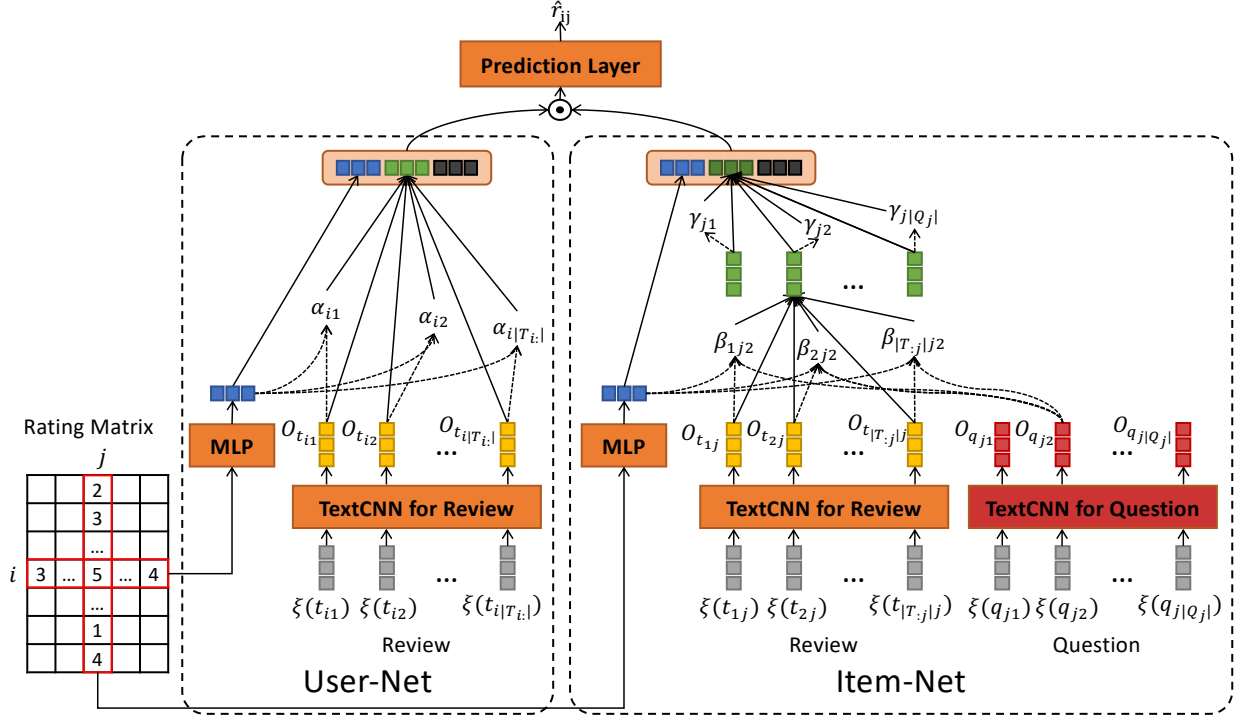


Figure 4.2: QUESTER model

The overall architecture of our proposed QUESTER model is shown in Figure 4.2. Below we describe its various components.

Text Encoder. We use a widely adopted CNN text processor [8, 51, 99], named TEXTCNN, for encoding to extract semantic features of text. TEXTCNN consists of a Convolutional Neural Network (CNN) followed by max pooling and a fully connected layer. Particularly, we have a word embedding function $\xi : M \rightarrow \mathbb{R}^D$ to map each word in the text t into a D -dimensional vector, forming an embedded matrix $\xi(t)$ with fixed length W (padded zero for text with length $< W$). Following this embedding layer is a convolutional layer with m neurons, each associated with a filter $F \in \mathbb{R}^{w \times D}$, each k^{th} neuron produces features by applying convolution operator on the embedded matrix $\xi(t)$:

$$z_k = \text{ReLU}(\xi(t) * F_k + b_z) \quad (4.1)$$

$ReLU(x) = \max(x, 0)$ is a nonlinear activation function and $*$ is the convolution operation. With sliding window w , the produced features would be $z_1, z_2, \dots, z_k^{W-w+1}$, which are passed to a max pooling to capture the most important features having highest values, which is defined as:

$$o_k = \max(z_1, z_2, \dots, z_k^{W-w+1}) \quad (4.2)$$

We get the final output of the convolutional layer by concatenating all output from m neurons, $O = [o_1, o_2, \dots, o_m]$. A simple approach to get the final representation of the input text t is to pass O into a fully connected layer as follows:

$$X = WO + b \quad (4.3)$$

Rating Encoder. Ratings are explicit features provided by users to indicate their interest on given items. The user ratings $r_{i:}$ form a rating pattern for user i , and the item ratings $r_{:j}$ form a rating pattern for item j . A reasonable choice is to use a multi-layer perceptron (MLP) network to learn the representation for the rating pattern [51]. Specifically,

$$\begin{aligned} h_{i1} &= \tanh(W_{r_{i:1}}r_{i:} + b_{r_{i:1}}) \\ h_{i2} &= \tanh(W_{r_{i:2}}h_{i1} + b_{r_{i:2}}) \\ &\dots \\ u_i &= \tanh(W_{r_{i:k}}h_{i(k-1)} + b_{r_{i:k}}) \end{aligned} \quad (4.4)$$

The output u_i is the final rating-based representation of user i , h_{ik} is the output hidden representation at layer k of the MLP. Similarly, we can also get the rating-based representation p_j of product j from its input ratings $r_{:j}$ in similar manner. We use \tanh as activation function to project the learned rating-based representation into the same range of text-based representations that will be discussed in the following paragraphs.

User Attention-Based Review Pooling. Equation 4.3 presumes that the contribution of each review is the same towards the final representation. The importance of each individual review contributing to user final representation is learnt as follows:

$$\rho_{ij} = \tanh(W_{O_t}(O_{t_{ij}} \odot u_i) + b_\rho) \quad (4.5a)$$

$$\theta_{ij} = W_\rho \rho_{ij} + b_\theta \quad (4.5b)$$

$$\alpha_{ij} = \frac{\exp(\theta_{ij})}{\sum_j \exp(\theta_{ij})} \quad (4.5c)$$

where \odot is element-wise multiplication operator, u_i is the rating-based representation of the user i , $O_{t_{ij}}$ is the feature vector extracted from review text t_{ij} by TEXTCNN, α_{ij} is the normalized attention score of the review t_{ij} , which can be interpreted as the contribution of that review to the feature profile O_i of user i , aggregating as follows:

$$O_i = \sum_j \alpha_{ij} O_{t_{ij}} \quad (4.6)$$

The final representation of user i is computed as follows:

$$X_i = W_{O_i} O_i + b_X \quad (4.7)$$

Item Question-Attentive Review-Level Explanations. Of particular importance is our modeling of product questions. A naive approach to model question on item side is to apply similar approach of modeling reviews. However, the connection between reviews and questions would have been overlooked. Here we presume that a product review may contain information that could be relevant to a question. We aggregate another attention layer based on item questions that help us to incorporate reviews based on their contribution towards item questions.

In particular, let $O_{t_{ij}}$ be the review encoding and $O_{q_{jk}}$ be the question² encoding of the prod-

²Each question is presumed to be accompanied by answer(s).

uct j . With respect to each question representation $O_{q_{jk}}$, we learn the attention weights β_{ijk} for review representation $O_{t_{ij}}$ by projecting both question and review representation onto an attention space followed by a non-linear activation function; the outputs are ϕ_{jk} and ρ'_{ij} respectively. We use tanh activation function to scale $O_{q_{jk}}$ and $O_{t_{ij}}$ to the same range of values, so that neither component dominates the other. To learn the question-specific attention weight of a review, we let the question projection ϕ_{jk} interact with the review projection ρ'_{ij} in two ways: element-wise multiplication and summation. The learned vector V plays the role of global attention context. This produces an attention value η_{ijk} , which is normalized using softmax to obtain β_{ijk} :

$$\phi_{jk} = \tanh(W_{O_q} O_{q_{jk}} + b_\phi) \quad (4.8a)$$

$$\rho'_{ij} = \tanh(W_{O_t} (O_{t_{ij}} \odot p_j) + b_{\rho'}) \quad (4.8b)$$

$$\eta_{ijk} = V^T (\phi_{jk} \odot \rho'_{ij} + \rho'_{ij}) \quad (4.8c)$$

$$\beta_{ijk} = \frac{\exp(\eta_{ijk})}{\sum_i \exp(\eta_{ijk})} \quad (4.8d)$$

Using the question-specific attention weights β_{ijk} , we aggregate the review representations $O_{t_{ij}}$'s into a question-specific representation d_{jk} as follows.

$$d_{jk} = \sum_i \beta_{ijk} O_{t_{ij}} \quad (4.9)$$

For a document (a product question with all of its reviews), we apply this attention mechanism for every product question, yielding a set of question-specific document representations d_{jk} , $k \in [1, |Q_j|]$. All the d_{jk} 's need to be aggregated into the final document representation O_j before incorporating to product representation. Thus, we seek to learn the importance weight

γ_{jk} , signifying how each question-specific representation d_{jk} would contribute to O_j .

$$\kappa_{jk} = K^T \tanh(W_{d_{jk}} d_{jk} + b_\kappa) \quad (4.10a)$$

$$\gamma_{jk} = \frac{\exp(\kappa_{jk})}{\sum_k \exp(\kappa_{jk})} \quad (4.10b)$$

Question-specific representation d_{jk} is projected into attention space through a layer of neurons with non-linear activation function \tanh . The scalar κ_{jk} indicates the importance of d_{jk} , obtained by multiplying with global attention context vector K (randomly initialized and learned during training). The representation d_{jk} 's due to the various questions are aggregated into the final product representation O_j using soft attention pooling with attention weight γ_{jk} 's.

$$O_j = \sum_k \gamma_{jk} d_{jk} \quad (4.11a)$$

$$Y_j = W_{O_j} O_j + b_Y \quad (4.11b)$$

Prediction Layer. The latent factors of user i and product j are mapped to a shared hidden space as follows:

$$h_{ij} = [u_i; X_i; \zeta_u(i)] \odot [p_j; Y_j; \zeta_p(j)] \quad (4.12)$$

where $\zeta_u(\cdot)$ and $\zeta_p(\cdot)$ are embedding function to map each user and each product into their embedding space respectively, X_i is user preferences and Y_j is item features obtained from user reviews and product reviews and questions, $[u_i; X_i; \zeta_u(i)]$ is the concatenation of user rating-based representation u_i , user text attention review pooling X_i , and user i embedding $\zeta_u(i)$. The final rating prediction is computed as follows:

$$\hat{r}_{ij} = W^T h_{ij} + b_i + b_j + \mu \quad (4.13)$$

Learning. Similar to prior works on rating prediction task [8, 51, 73], which is a regression problem, we adopt the squared loss function:

$$\mathcal{L} = \sum_{i,j \in \Omega} (\hat{r}_{ij} - r_{ij})^2 \quad (4.14)$$

Where Ω denotes the set of all training instances, r_{ij} is the ground truth rating that user i assigned on product j .

The most important question L is selected by $L = \text{argmax}_k(\gamma_{jk})$ and the most useful review is selected by $\text{argmax}_i(\beta_{ijL})$. We use the selected question with its answer and the selected review collectively as explanation for a given recommendation.

A limitation of relying only on questions found within a product is that product features may not be captured completely, because some products do not have sufficient questions to cover all its important aspects. As a result, an important review may be overlooked because it does not correspond to any question. To address this limitation, in addition to the questions found in a product, we include one more global ‘‘General Question’’, which allows those important reviews to still be aligned. This additional question plays the role of ‘‘global’’ aspect, and also helps our model to potentially generalize to product without questions.

4.2 Experiments

As this work is primarily about recommendation explanations, rather than rating prediction per se, and the two objectives are not necessarily directional equivalent, our orientation is to improve explanations while maintaining parity in accuracy performance. In particular, our core contribution is in incorporating question and answer or QA for review-level explanation. The experimental objectives revolve around the utility of QA as part of explanation, the effectiveness of QA to aid the selection of review-level explanation, and the alignment of QA and review that are part of an explanation.

Dataset	#Item	#User	#Review (Rating)	#Question	#Item with Question #Item
Home	28,169	66,295	549,895	368,904	0.3193
Health	18,464	38,416	344,888	105,814	0.1731
Sport	18,301	35,447	295,074	123,119	0.1940
Toy	11,870	19,322	166,821	35,520	0.1463
Grocery	8,690	14,632	150,802	18,134	0.1301
Baby	7,039	19,418	160,521	32,507	0.1301
Office	2,414	4,892	53,143	68,864	0.4544
Automotive	1,810	2,892	20,203	40,477	0.3470
Patio	951	1,667	13,133	22,454	0.3049
Musical	893	1,416	10,163	22,409	0.5622

Table 4.2: Data statistics

Datasets. Towards reproducibility, we work with publicly available sources. While QA is a feature on many platforms, not many such datasets have both reviews and QA information. One that does is the Amazon Product Review Dataset³ [22]. We experiment on ten product categories from this source as separate instances. These categories are selected for significant availability of QA information. Consistent performance across multiple categories with different statistics bolster the analysis. Table 4.2 summarizes basic statistics of the ten datasets.

For greater coverage, we collect item questions and acquire their helpful voting scores from the Amazon.com website. Too short reviews (less than 3 words), users and items with fewer than five reviews are filtered out. For each question, we also include one answer (the earliest that appears in the data) as frequently answers are similar. To aggregate overlapping questions, we cluster questions in each category with KMeans, keeping questions from big clusters which cover 80% of questions. For smaller clusters, we keep the nearest question to each cluster centroid and combine them into a single text, called General Question (all products have this by default). This is used solely for modeling to generalize to items without questions, but would not be used as a recommendation explanation.

Baselines. We evaluate our proposed QUESTER against the following baselines in terms of useful review and QA selection. Comparisons between methods are tested with one-tailed paired-

³<http://jmcauley.ucsd.edu/data/amazon/>

sample Student’s t-test at 0.05 level.

- **HRDR** [51] uses attention mechanism with the rating-based representation as features to weight the contribution of each individual review toward user/item final representation.
- **NARRE** [8] learns to predict ratings and the usefulness of each reviews by applying attention mechanism for reviews on users/items embedding.
- **HFT** [55] models the latent factors from user or item reviews by employing topic distributions. In this work, we employ item reviews and applied their proposed usefulness review retrieval approach for selecting useful reviews. The number of topics is $K = 50$.

Note that our key distinction from the above mentioned baselines is that we further incorporate product questions. As there is no prior work on predicting ratings along with selecting useful question, when the evaluative task is to look into selecting questions (question retrieval and question similarity tasks, see Section 4.2.1 and Section 4.2.3), we would apply similar approach for each baseline such that item text will be item questions instead of item reviews.

Training Details. Each item’s reviews are split randomly into train, validation, and test with ratio 0.8 : 0.1 : 0.1. Unknown users are excluded from validation and test sets. We employ the pretrained word embeddings from GloVe [65] to initialize the text embedding matrix with dimensionality of 100 in which the embedding matrix is shared for both reviews and questions. We use separate TEXTCNN for user reviews, item reviews, and item QAs. Max text length W is 128, the number of neurons in convolutional layer m is 64, the window size w is 3. The latent factor number was tested in $k \in \{8, 16, 32, 64\}$. After tuning, we set $k = 8$ for memory efficiency as using larger k does not improve the performance significantly. Dropout ratio is 0.5 as in [8]. We apply 3-layers MLP for rating-based representation modeling as in [51], with the number of neural units in hidden layers to be $\{|l|, 128, 64, m\}$ where $|l|$ is the number of items (resp. number of users) for user-net (resp. item-net). Using Adam optimizer [30] with an initial learning rate of 10^{-3} and mini-batch size of 64, we see models tend to converge before 20 epochs.

We set a maximum of 20 epochs and report the test result from the best performing model (MSE) on validation, a uniform practice across methods.

Brief Comment on Running Time. Our focus in this work is recommendation explanation, rather than computational efficiency. The models can be run offline. For a sense of the running times, our model takes between 5 minutes on the Musical category to 5 hours on the Home category on AMD EPYC 7742 64-Core Processor and NVIDIA Quadro RTX 8000. The running times of the baselines are generally in the same ballpark.

4.2.1 Question and Review Alignment

Our proposed recommendation explanation consists of a question-and-answer (QA) and a review. Ideally, these two components, QA on one hand, and review on the other hand, are well-aligned for a more coherent explanation. We measure this alignment using ROUGE [46] and METEOR [1], two well-known metrics for text matching and text summarization. To cater to words as well as phrases, we report F-Measure of ROUGE-1 (R-1) measuring the overlapping unigrams, ROUGE-2 (R-2) measuring the overlapping bigrams, and ROUGE-L (R-L) measuring the longest common subsequence between the reference summary and evaluated summary. We compute ROUGE and METEOR scores for the top-1 selected question and review and report them in Table 4.3.

The results show that the proposed QUESTER consistently outperforms the baselines significantly across virtually all the datasets. This shows QUESTER’s QAs and reviews that are part of a collective explanation are better-aligned with each other, as compared to the respective pairings identified by the baselines. Note that HRDR, NARRE, and HFT had been designed solely to select helpful reviews. To be able to compare with these models, we ran each model twice, once with reviews and another time replacing item reviews with QA’s. This approach essentially treats review and question in a disjoint manner, which contributes to why they are underper-

Data	Model	R-1	R-2	R-L	METEOR
Home	QUESTER	15.73 [§]	0.93 [§]	7.91 [§]	10.27 [§]
	HRDR	14.71	0.74	6.91	8.07
	NARRE	14.70	0.72	6.75	7.72
	HFT	13.53	0.65	6.38	7.49
Health	QUESTER	20.31 [§]	1.73 [§]	8.68 [§]	11.20 [§]
	HRDR	19.77	1.60	7.63	8.93
	NARRE	18.09	1.33	6.35	7.32
	HFT	17.13	1.27	6.57	7.88
Sport	QUESTER	15.92 [§]	0.80 [§]	7.83 [§]	10.05 [§]
	HRDR	14.96	0.60	6.72	7.77
	NARRE	14.15	0.51	5.86	6.51
	HFT	13.86	0.56	6.09	7.27
Toy	QUESTER	16.14 [§]	1.30 [§]	8.39 [§]	10.57 [§]
	HRDR	15.25	1.09	7.24	8.20
	NARRE	14.90	0.99	6.82	7.52
	HFT	14.03	0.96	6.51	7.40
Grocery	QUESTER	17.29 [§]	0.79 [§]	7.51 [§]	9.09 [§]
	HRDR	16.77	0.69	6.77	7.56
	NARRE	15.03	0.57	5.43	5.78
	HFT	14.70	0.58	5.71	6.45
Baby	QUESTER	19.55 [§]	1.36 [§]	8.45 [§]	12.00 [§]
	HRDR	18.98	1.24	7.91	10.70
	NARRE	17.60	1.03	6.79	8.52
	HFT	15.94	0.87	6.14	7.62
Office	QUESTER	18.11 [§]	1.03 [§]	7.84 [§]	13.05 [§]
	HRDR	17.64	0.77	7.38	11.27
	NARRE	17.22	0.71	6.77	9.20
	HFT	15.02	0.58	6.30	8.92
Automotive	QUESTER	18.29 [§]	1.22 [§]	8.06 [§]	10.86 [§]
	HRDR	17.57	1.19	7.57	10.28
	NARRE	16.33	0.91	6.16	7.35
	HFT	15.45	0.86	6.53	8.27
Patio	QUESTER	19.25 [§]	1.87 [§]	9.17 [§]	13.74 [§]
	HRDR	18.35	1.71	8.69	12.92
	NARRE	16.90	1.30	7.11	9.39
	HFT	15.78	1.26	7.29	10.60
Musical	QUESTER	16.57 [§]	0.94 [§]	7.54 [§]	11.64 [§]
	HRDR	15.15	0.72	6.61	9.66
	NARRE	15.53	0.75	6.86	8.35
	HFT	12.94	0.59	5.88	8.72

[§] denotes statistically significant improvements. Highest values are in **bold**.

Table 4.3: Performance in question and review alignment

forming as compared to our proposed QUESTER that jointly selects review and question that are well-aligned with each other.

4.2.2 Review-Level Explanation

Here we assess whether incorporating questions would help in selecting reviews for the explanation. We take reviews that have the greatest positive helpfulness voting scores on every product to be the ground truth to study the performance of selecting useful reviews. We use Precision

Data	Model	Review-Level Explanation						
		Prec@5	Rec@5	F1@5	R-1	R-2	R-L	METEOR
Home	QUESTER	0.147 [§]	0.643 [§]	0.234 [§]	36.35 [§]	20.41 [§]	26.56 [§]	31.25 [§]
	HRDR	0.133	0.574	0.211	30.94	15.16	21.21	24.24
	NARRE	0.134	0.580	0.213	29.70	13.94	19.98	23.69
	HFT	0.140	0.611	0.223	28.76	14.21	19.85	23.23
Health	QUESTER	0.152 [§]	0.648 [§]	0.241 [§]	36.23 [§]	20.84 [§]	26.76 [§]	32.52 [§]
	HRDR	0.138	0.581	0.217	30.76	15.14	21.15	25.62
	NARRE	0.134	0.560	0.210	26.14	11.28	16.96	20.31
	HFT	0.149	0.634	0.235	28.83	14.85	20.32	23.98
Sport	QUESTER	0.159 [§]	0.671 [§]	0.251 [§]	37.24 [§]	22.01 [§]	27.86 [§]	33.50 [§]
	HRDR	0.146	0.611	0.230	30.87	15.32	21.34	26.15
	NARRE	0.140	0.583	0.220	26.50	11.44	17.16	20.43
	HFT	0.155	0.654	0.245	29.80	15.70	21.14	24.92
Toy	QUESTER	0.160 [§]	0.691 [§]	0.254 [§]	39.31 [§]	23.23 [§]	29.11 [§]	34.68 [§]
	HRDR	0.143	0.611	0.226	31.75	15.22	21.19	25.95
	NARRE	0.141	0.605	0.224	29.20	13.04	18.84	22.81
	HFT	0.150	0.645	0.238	30.18	15.48	20.82	24.43
Grocery	QUESTER	0.167 [§]	0.702 [§]	0.263 [§]	37.10 [§]	21.74 [§]	27.75 [§]	33.89 [§]
	HRDR	0.157	0.660	0.247	33.37	17.63	23.79	29.26
	NARRE	0.150	0.626	0.235	27.74	12.55	18.39	22.06
	HFT	0.162	0.681	0.255	30.46	16.10	21.70	25.76
Baby	QUESTER	0.141 [§]	0.598 [§]	0.223 [§]	36.58 [§]	18.85 [§]	25.38 [§]	32.03 [§]
	HRDR	0.130	0.549	0.206	34.81	17.23	23.58	29.70
	NARRE	0.119	0.495	0.187	28.53	11.35	17.50	21.50
	HFT	0.129	0.542	0.203	27.86	12.50	18.02	21.48
Office	QUESTER	0.149 [§]	0.607 [§]	0.228 [§]	37.40 [§]	20.62 [§]	26.36 [§]	34.01 [§]
	HRDR	0.134	0.545	0.205	33.80	16.49	22.40	29.05
	NARRE	0.124	0.500	0.189	26.28	9.74	15.20	19.69
	HFT	0.125	0.513	0.193	27.39	12.44	17.46	21.52
Automotive	QUESTER	0.172	0.733	0.272	37.12	22.41	28.28	33.72
	HRDR	0.175	0.739	0.276	36.34	21.33	27.35	32.70
	NARRE	0.156	0.651	0.245	26.89	12.09	17.70	21.28
	HFT	0.167	0.709	0.264	29.84	15.81	21.33	25.11
Patio	QUESTER	0.167	0.698	0.258	38.42 [§]	22.05 [§]	27.86 [§]	34.66 [§]
	HRDR	0.165	0.676	0.252	34.81	17.60	23.65	30.78
	NARRE	0.152	0.629	0.233	28.34	11.75	17.34	22.46
	HFT	0.167	0.696	0.257	32.77	17.78	23.10	27.26
Musical	QUESTER	0.179 [§]	0.763 [§]	0.284 [§]	37.29	21.78	27.58	35.11
	HRDR	0.173	0.733	0.274	35.81	20.59	26.16	32.84
	NARRE	0.161	0.677	0.255	27.44	12.08	17.71	21.65
	HFT	0.173	0.730	0.274	30.86	16.75	21.91	26.66

[§] denotes statistically significant improvements. Highest values are in **bold**.

Table 4.4: Performance in Review-Level Explanation task

at 5 (Prec@5), Recall at 5 (Rec@5), and F1@5 as evaluation. As reported in Table 4.4, our proposed QUESTER is the better-performing method overall. Its outperformance over baseline models is statistically significant in the vast majority of the cases. For Automotive and Patio categories, QUESTER still outperforms NARRE (on Automotive and Patio categories) and HFT (on Automotive category) significantly.

To further assess the quality of top-ranked reviews against top-rated helpful reviews, we again use ROUGE and METEOR as metrics. The results in Table 4.4 consistently show that our proposed QUESTER outperforms all baseline models significantly in all measurements, i.e.,

Data	Model	Question-Level Explanation						
		Prec@5	Rec@5	F1@5	R-1	R-2	R-L	METEOR
Home	QUESTER	0.086 [§]	0.325 [§]	0.130 [§]	23.07 [§]	9.36 [§]	16.10 [§]	19.67 [§]
	HRDR	0.082	0.309	0.125	19.70	7.13	12.98	16.13
	NARRE	0.083	0.309	0.125	19.05	6.40	12.13	15.46
	HFT	0.082	0.312	0.125	18.40	7.43	13.19	15.00
Health	QUESTER	0.097 [§]	0.378 [§]	0.150 [§]	21.98 [§]	8.92 [§]	15.07 [§]	20.10 [§]
	HRDR	0.089	0.345	0.137	16.75	3.01	7.52	14.23
	NARRE	0.091	0.351	0.139	19.24	7.84	13.16	15.94
	HFT	0.091	0.353	0.140	18.82	8.58	13.97	16.15
Sport	QUESTER	0.093 [§]	0.360 [§]	0.143 [§]	23.15 [§]	9.86 [§]	16.22 [§]	20.87 [§]
	HRDR	0.085	0.329	0.131	15.21	3.37	7.94	12.97
	NARRE	0.088	0.336	0.135	18.31	6.25	11.71	15.28
	HFT	0.091	0.346	0.139	20.01	9.02	14.91	16.92
Toy	QUESTER	0.110 [§]	0.411 [§]	0.167 [§]	23.61 [§]	11.03	17.15	23.29 [§]
	HRDR	0.105	0.392	0.160	15.95	4.45	8.95	15.03
	NARRE	0.107	0.396	0.162	21.48	10.35	15.69	20.25
	HFT	0.109	0.395	0.164	22.10	11.96 [§]	17.16 [§]	20.65
Grocery	QUESTER	0.112 [§]	0.467 [§]	0.176 [§]	23.11	11.01	16.90	20.80
	HRDR	0.098	0.405	0.154	17.10	3.96	8.43	15.36
	NARRE	0.102	0.421	0.161	23.51 [§]	12.07 [§]	17.11 [§]	21.05 [§]
	HFT	0.106	0.437	0.166	20.05	9.81	15.02	16.69
Baby	QUESTER	0.081	0.309	0.124	22.40 [§]	8.46	15.38 [§]	20.51 [§]
	HRDR	0.081	0.305	0.123	17.82	4.56	10.19	16.42
	NARRE	0.089	0.341 [§]	0.136 [§]	21.59	8.56	14.79	19.03
	HFT	0.090 [§]	0.333	0.136 [§]	19.88	9.24 [§]	15.31	16.46
Office	QUESTER	0.074	0.291	0.114	20.57 [§]	6.40	12.86 [§]	16.68 [§]
	HRDR	0.076 [§]	0.295 [§]	0.117 [§]	16.76	2.93	8.16	12.40
	NARRE	0.074	0.285	0.113	16.69	3.24	8.45	12.56
	HFT	0.076 [§]	0.294	0.116	17.91	6.73 [§]	12.42	13.64
Automotive	QUESTER	0.065	0.252	0.099	20.35 [§]	6.02 [§]	12.83 [§]	16.84 [§]
	HRDR	0.067 [§]	0.279 [§]	0.105 [§]	17.99	4.95	11.10	13.60
	NARRE	0.063	0.253	0.098	18.46	4.61	10.28	14.22
	HFT	0.060	0.245	0.093	16.48	5.26	11.18	12.32
Patio	QUESTER	0.060 [§]	0.250 [§]	0.094 [§]	19.22 [§]	4.85 [§]	11.85 [§]	14.87 [§]
	HRDR	0.052	0.196	0.079	16.04	3.01	8.62	11.50
	NARRE	0.055	0.210	0.084	14.17	2.02	6.59	10.37
	HFT	0.055	0.206	0.083	15.32	4.65	10.25	9.80
Musical	QUESTER	0.082	0.333	0.128	22.93 [§]	8.82 [§]	14.81 [§]	19.79 [§]
	HRDR	0.085 [§]	0.335 [§]	0.131 [§]	17.19	3.67	9.35	13.22
	NARRE	0.078	0.312	0.121	21.90	6.71	13.30	18.25
	HFT	0.082	0.328	0.127	17.22	5.57	11.35	12.27

[§] denotes statistically significant improvements. Highest values are in **bold**.

Table 4.5: Performance in Question-Level Explanation task

the top-ranked reviews from QUESTER are more similar to the top-rated helpful reviews than those of HRDR, NARRE, and HFT. Overall, in addition to the reviews, our QUESTER uses additional product QA, achieving better results than the baseline methods those only use reviews as additional data, suggesting that using QA aids in selecting more useful reviews.

Data	HFT	NARRE	HRDR	QUESTER
Home	1.2796	1.2654	1.2666	1.2661
Health	1.2628	1.2853	1.2875	1.2862
Sport	1.0231	1.0054	1.0055	1.0046
Toy	0.9129	0.9960	0.9980	0.9955
Grocery	1.1992	1.1988	1.1983	1.1998
Baby	1.3698	1.3621	1.3650	1.3609
Office	0.8943	0.9248	0.9259	0.9249
Automotive	0.9574	0.9248	0.9248	0.9256
Patio	1.1153	1.1537	1.1549	1.1585
Musical	1.0627	0.8889	0.8861	0.8788
<i>Average</i>	1.1077	1.1005	1.1013	1.1001

Table 4.6: Rating prediction performance: Mean Square Error

4.2.3 Question-Level Explanation

The novelty of the proposed QUESTER is in producing question-level explanation along with review-level explanation. We conduct a homologous quantitative evaluation as Review-Level Explanation above, but now with question votes as ground-truth and measure Prec@5, Rec@5, and F1@5. In addition, we measure the similarity between question generated by QUESTER and top voted useful question using ROUGE and METEOR, the first answer of each question is concatenated as a part of the question text for evaluation. As shown in Table 4.5, QUESTER is competitive throughout. In many cases it shows better results than the baselines, and frequently in a statistically significant manner. Notably, a baseline never beats the proposed method in a statistically significant manner.

4.2.4 Rating Prediction

As previously established, our main focus in this work is on recommendation explanations, with an eye on improving the selection of reviews and incorporating questions in that endeavour. Nevertheless, while recommendation accuracy is not the main focus, we find that QUESTER still maintains parity in this regard with the other methods.

We report the average of *Mean Square Error* (MSE) averaged across users on each category

in Table 4.6. While the performances of various methods vary slightly across categories, the average MSE across categories (the last row) for QUESTER is slightly lower (better). Our proposed QUESTER achieves comparable results when compared to the neural models HRDR and NARRE. HFT that is based on graphical model varies from the neural models. Depending on the reported domain, it is lower in some cases and higher in others. Such variation in performance between simpler and more complex models using neural networks in term of rating predictions is expected and has also been reported in [69].

In any case, as we see from the previous experiments as well, QUESTER stands out in having the better review-level and question-level explanations, which are the main focal points of this work.

4.2.5 Case Studies

To investigate the usefulness of the recommendation explanation consisting of a QA as well as a review, we show a few case studies that benchmarks QUESTER to the most voted question and the most voted review.

Figure 4.3 shows two sets of explanations for a sanding pad product of *Meguiar's* brand. The first set (in grey box, above) comprise a QA and a review based on Top_Rated_Useful votes. The second set (in green box, below) comprise those selected by our QUESTER. While both QUESTER and Top_Rated_Useful provide useful information about the product, QUESTER's explanation is notable in two respects. For one, QUESTER's question with its answer is more aligned with its review than those of Top_Rated_Useful, ROUGE-L F-Measure for QUESTER and Top_Rated_Useful are 10.61 and 8.37 respectively. For another, Top_Rated_Useful is based on explicit votes, which are not found on many products and therefore not universally available or applicable.

Figure 4.4 shows explanation for a breast pump product of *Medela* brand. Both QUESTER and Top_Rated_Useful provide further useful information about the product. QUESTER's ques-



Asin: B0009IQZ2K
Title: Meguiar's E7200 Mirror Glaze High-Tech Backing Pad

Top Rated Useful Question: What grit is this?
Answer: There is no grit. It is a flexible sanding pad that you wrap your wet/dry sandpaper around to allow for easier sanding.

Top Rated Useful Review: I'm not a Meguiar's fan when it comes to their polishes and cleaning supplies, but this pad seems to work well. I have better control of it when it's cut in half width wise, which gives me two square blocks.

QUESTER Question: Is this soft and flexible enough to be used for fine wet sanding? I basically need something that bends to the contours.
Answer: This is soft and flexible but I don't know if it is flexible enough to be able to bend to contours while you are placing even pressure across the entire sponge. If you were to purposefully distribute weight to the correct areas of the sponge then yes, but expecting the sponge to form fit to the area being sanded is not feasible.
QUESTER Review: There's really not a lot to it. It's just a small 5-1/2" long X 2-1/2" wide foam block, but it worked fine as a backing pad along with 2000 grit paper for wet sanding clear coat before polishing. It seems to be holding up okay to use, so for what little it cost I think it was worth getting.

Figure 4.3: Example explanation: Meguiar's Sanding Pad (explanation by Top_Rated_Useful is in grey, that by QUESTER is in green)

tion with its answer is considered more aligned with its review than those of Top_Rated_Useful, ROUGE-L F-Measures are 12.59 and 9.02 respectively.

In turn, Figure 4.5 shows explanation for a guitar rest. Notably, the pairing by Top_Rated_Useful are not so coherent, with the QA discusses its use for guitars, while the review discusses its use for ukuleles. In contrast, both the QA and the review by QUESTER focus on the key issue of how well the item could hold a guitar in rest. QUESTER's QA is more aligned with its review than those of Top_Rated_Useful, ROUGE-L F-Measures are 14.71 and 6.64 respectively.

4.2.6 User Studies

To evaluate the quality of questions and reviews selected by QUESTER and Top_Rated_Useful (based on user votes on Amazon.com), we conduct a couple of user studies.

Reviews vs. QAs. In the first study, we seek to investigate whether users find questions and reviews helpful as part of a recommendation explanation. We conduct user studies concerning 30 examples (3 products from each category). We split these examples into 3 surveys, each



Asin: B0006HBS1M

Title: Medela, Harmony Breast Pump, Manual Breast Pump, Portable Pump, 2-Phase Expression Technology, Ergonomic Swivel Handle, Easy to Control Vacuum, Designed for Occasional Use

Top Rated Useful Question: Does this use the same bottles as the Medela electric pumps?

Answer: yes. same bottles come with electric pump, manual pump and cooler bag

Top Rated Useful Review: I love this pump. I was told by my doctor to pump to stimulate my milk supply and this totally helped. It is worth the money it costs. Think of it as an investment in your baby. Plus if you have more than 1 child you can use it again.

QUESTER Question: What size O-ring can be used to replace the original?

Answer: I found just a generic one in amazon and it worked.

QUESTER Review: I love this. Saved me. Take it with me to work and pump 2-3 times, get about 8-10 oz all together. Pump for 10 min each time. Easy to clean. I also have Pump in Style and cannot get nothing. Went to lactation consultant 4 times and finally took this out of despair and it worked!

Figure 4.4: Example explanation: Medela’s Breast Pump (explanation by Top_Rated_Useful is in grey, that by QUESTER is in green)

containing 10 examples of different domains which are generated by QUESTER. Each survey is done by 5 annotators, for a total of 15 annotators who are neither the authors nor having any knowledge of the objective of the study. Each product is presented with both question and review ordering randomly (review and question can be either group A or group B). We ask annotators to assess the pairwise quality with four options: (i) A is more useful than B; (ii) B is more useful than A; (iii) A and B are almost the same, both useful; (iv) A and B are almost the same, both useless. The Fleiss’ Kappa [35] score for consistency for categorical ratings, $\kappa = 0.2955$, indicates fair agreement.

Pairwise evaluation results are shown in Figure 4.6. As the key proposal is to have both review and question be part of an expanded explanation, it is gratifying that the most popular option is that both are useful, attaining 39.3%. While the percentage that finds reviews more useful is slightly higher than the percentage that finds questions more useful, this is less important as we are not seeking to replace reviews with questions. Excluding “both useless”, 96% find at least one useful. We repeat the same study with explanations coming from Top_Rated_Useful and the conclusion still holds, i.e., the most popular option is that both reviews and QA are useful.

QuestER vs. Top_Rated_Useful. In the second user study, we would like to investigate the quality of the proposed *combined* explanation form consisting of a QA and a review. With



Asin: B004N0MKN8
Title: Planet Waves Guitar Rest

Top Rated Useful Question: What is the response to the numerous customer reviews that say that the thing keeps falling off unless the guitar is resting against it?

Answer: It does tend to fall off, like you say, but it really is great to lean the guitar on. Otherwise the guitar just falls over. Pick your poison! Sorry

Top Rated Useful Review: The Planet Waves Guitar Rest works for ukuleles! I just got one, and have used it for a few days, and it's the bomb! I can set my little ukuleles down now without fear of falling over. This product is a rubber disc with small "arms" in a gentle curve that nestles against the edge of any surface, and you can set your instrument against it, and voila, it doesn't fall over! Here at home, I use it on the second shelf of a bookcase, and my concert sized ukulele fits like a glove, heel on carpet, neck in Guitar Rest. I'm going to buy a couple more for my ukulele cases, because I can use them at one of my uke parties. If one sets a tiny ukulele on the floor, for instance, to take a whizz, they're just small enough to go unseen and have someone step on them. Here, I just find a spot near wherever I'm sitting, and it becomes my "lean" spot, and I can even set my beer can on the round part on the back! Coaster uke/guitar holder. It's quite immovable once it has some weight against it from the instrument. I could carry a metal stand with me, but it wouldn't fit in my ukulele case--this Planet Waves product does. A winner.

QUESTER Question: Will this guitar rest work on a round table top?

Answer: That depends entirely on the dimensions of the table. Take the guitar and see if it can lay flat across the table. If it does, then it will work just fine. If it goes off the end a little bit, it should still be fine.

QUESTER Review: I've had this thing for several weeks, and just now, when it fell off the table for the 100th time, I tossed it in the trash. The whole thing is one piece of soft floppy rubber, it's not stiff enough for the part that cradles the guitar neck, and it's not heavy enough to stay put. Even the force from the guitar neck makes it topple over. Unless you glue this thing to the table, or something like that, it's useless, even worse than useless, it's in the way.

Addendum: I raised the rating a bit, after hearing from the distributor/manufacturer ... at least these guys listen.

Figure 4.5: Example explanation: Planet Waves Guitar Rest (explanation by Top_Rated_Useful is in grey, that by QUESTER is in green)

the same set of examples and annotators, we split the examples into 3 other surveys, each containing 10 products from different categories. We present the explanations blindly by ordering survey's questions and explanations randomly (group A and group B are now either QUESTER or Top_Rated_Useful). We ask similar question as in the first study. Figure 4.7 shows the pairwise evaluation results between QUESTER and Top_Rated_Useful. The Fleiss' Kappa score is 0.217 indicating fair agreement. In summary, when combining question and review as explanation, the overall quality of both QUESTER and Top_Rated_Useful are useful (96.67%). Among those, question and review selected by QUESTER are considered to be slightly more useful (26.7%) than those of top rated useful (25.3%).

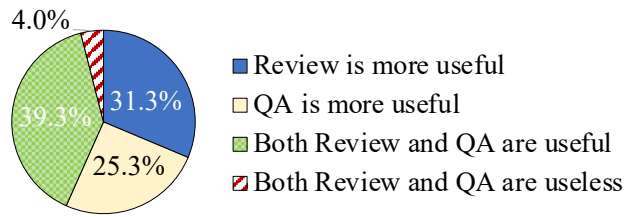


Figure 4.6: Review vs Question-Answer annotation results

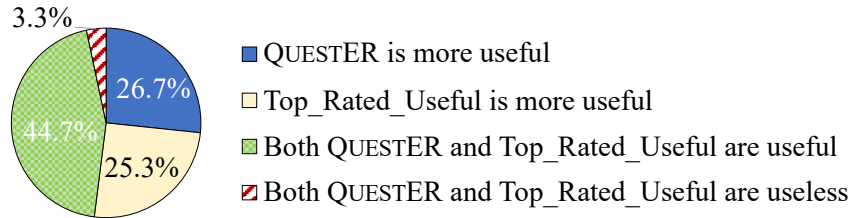


Figure 4.7: QUESTER vs Top_Rated_Useful annotation results

As important as the slight outperformance by QUESTER over Top_Rated_Useful, or perhaps more so is that QUESTER as a method is more widely applicable method. In contrast, Top_Rated_Useful relies on the existence of helpfulness votes, which are relatively rare, and therefore it stands more as a benchmark rather than a practical method for review and QA selection for explanation.

4.3 Summary

In this chapter, we describe QUESTER, a framework for incorporating question-answer pair or QA into review-based recommendation explanation. We model QA in an attention mechanism to identify more useful reviews. Through joint modeling, we can collectively form an explanation in terms of QA and review. Comprehensive experiments on various product categories show that the QA and the review that are part of a collective explanation are more coherent with each other than those pairings found by the baselines. Review-level and question-level explanations identified by QUESTER are also more consistent with top-rated ones based on helpfulness votes than those identified by the baselines. User studies further help to support that incorporating questions as part of a recommendation explanation is useful.

Part II

Comparative Recommendation

Explanations

Chapter 5

Explainable Recommendation with Comparative Constraints on Product Aspects

In recent past, we begin to see a build-up of interest in explainable recommendations [96]. The core of many models lies in anchoring the explanations on product aspects that have been mentioned by users in online reviews. For instance, a pioneering work EFM [96] produces an explanation in the form of “*You might be interested in [aspect], on which this product performs [well/poorly].*”. In turn, another well-known model MTER [85] produces an explanation in the form of “*Its [aspect] is [opinion phrase].*”. In these explanation templates, variables enclosed in square brackets are to be substituted with the relevant aspects, sentiments, or opinion phrases. Note how such explanations are *evaluative* by nature, assessing the quality of a single product in and of itself.

Comparative Explanation. We posit that users are inherently interested in choice-making, gaining information from relative comparisons. To this extent, binary sentiments are of insufficient precision in differentiating items (many of which may be equally ‘*positive*’, or ‘*negative*’). Neither is it easy to compare opinion phrases such as ‘*light*’ vs. ‘*portable*’, or ‘*affordable*’ vs.

'*value for money*'. Thus, we seek a comparative explanation for a recommended item, the manner of which is illustrated by the example below. Such comparative recommendation explanations may be used in addition or in place of evaluative ones.

```
[recommended item] is better at [an aspect] than  
[reference item], but worse at [another aspect].
```

One question is which items should serve as reference to a recommended item. There are several reasonable options. One could be a comparable substitute under consideration, e.g., a buyer of washing machines may wish to know how other washers in the market compare to the recommended one. Another could be a previously purchased item by the target user. Our focus is on the latter. For one reason, this is a comparison that the target user would likely find *understandable*, given her familiarity with the previous item. For another, the user may find it more *actionable* if it confers a perception of gain in improving upon one's past purchase. This would also characteristically be a *personalized* form of explanation, as it relates directly to the target user's past actions. The disadvantage lies in the classic cold-start scenario when the user has not previously purchased a similar product, in which case we could always fall back to an evaluative explanation for such scenarios.

Comparative Constraints. If we presuppose that a user generally tries to make better decisions over time by improving upon past purchases, then hypothetically the stereotypical users may already exhibit this behavior in their past purchase histories, at least to a certain extent. In other words, an explainable recommendation model that expects this behavior when learning the model has the potential of producing recommendations that are more reflective of user behavior and hopefully more accurate as well.

Therefore, we formulate comparative constraints relating historical purchases that can be incorporated into explainable recommendation models. Borrowing a terminology from the skyline literature [4, 50], we say that a product *y dominates* another product *x*, if the former is at least as

good as the latter in all aspects and better in at least one aspect. Now, it may not necessarily be the case that a later purchase y must always dominate a previous purchase x . On the other hand, it may be reasonable to assume that most of the time, y is not dominated by x . For example, while a user may go on to buy a cheaper model after finding that she does not need all the bells and whistles that come with a previously purchased more expensive model, it bears pointing out that the very fact that the later purchase is cheaper means that it is still superior in one way (i.e., value) and thus is not dominated by the earlier purchase.

In the explainable recommendation literature [85, 96], product aspects are often extracted from review text. When a later adoption is observed to be better at some aspect than a previous adoption by the same user, we formulate an aspect-level comparative constraint that seeks to preserve the same comparison in the modeled latent aspect sentiments. Moreover, our approach is to model these aspect-level comparative constraints as a framework, allowing specific instantiations built on two lines of explainable recommendation models, namely one that allows *subjective* aspect-level quality (user-specific) and another that accommodates *objective* aspect-level quality.

5.1 Product Ratings over Time

To gain insights in developing the comparative constraints using previous items as references, we conduct an empirical analysis of ratings that users give to products from a broad spectrum of categories over time. For this purpose, we use the public¹ Amazon.com dataset [22]. In this dataset, only the act of rating, rather than purchase, is visible. Note also that this discussion is of motivational rather than conclusive nature, and a fuller investigation of this hypothesis at aspect and category-levels will be done in Section 5.4.

Different Products Launched Across Time. First, we consider whether more recently launched products tend to induce greater ‘consumer satisfaction’, which may imply that gen-

¹<http://jmcauley.ucsd.edu/data/amazon/>

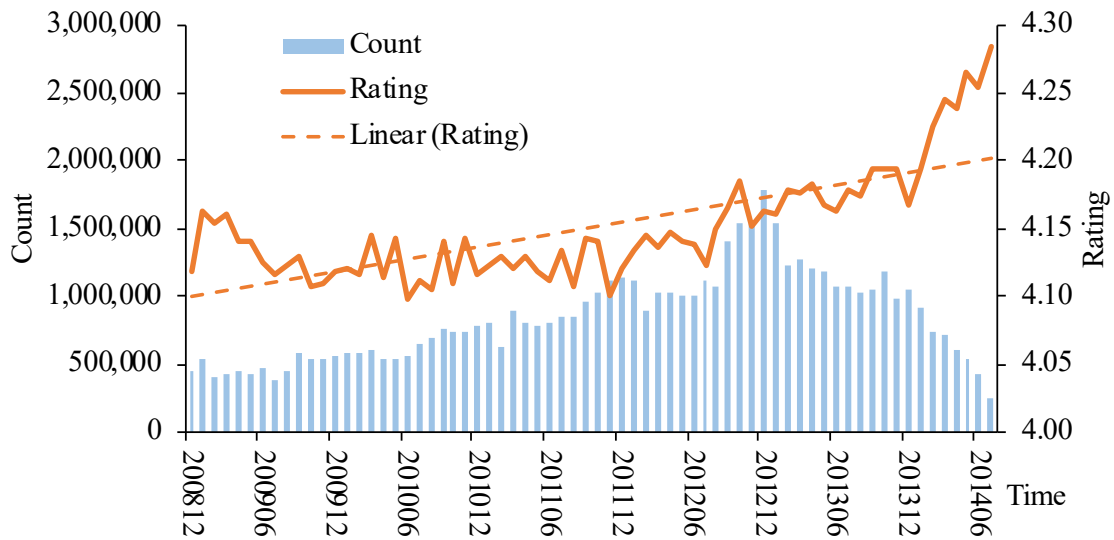


Figure 5.1: Average rating of products launched over time

erally speaking products tend to improve over time (with better features, ease of use, etc.). For this analysis, we associate every product with two attributes. One is its ‘estimated quality’ (i.e., average rating). The other is its ‘estimated launch time’ (i.e., time of its first review). These are but approximations, which may suffice as our intent is to study general trends involving many products.

We group all products ‘launched’ in the same month, and track their average ‘quality’ over time in Figure 5.1. It is evident from the line graph that, minor fluctuations notwithstanding, the general trend is that products launched later tend to have higher average rating over its ‘lifetime’. The dotted line provides the best-fitting line, which has a positive gradient. The basis for this analysis is a large number of reviews, as shown by the histogram in Figure 5.1. The rating count initially goes up, probably as the popularity of Amazon goes up. The later downturn is because products ‘launched’ in recent years have not reached their full potentials in terms of rating counts by the cut-off date in the dataset. Even the lowest bar of the histogram (July 2014) has been supported by 258K ratings.

Same Product Over Time. Second, we now group all the products together, but slice the set of ratings by the distance between the launch time and the time in which the rating is assigned.

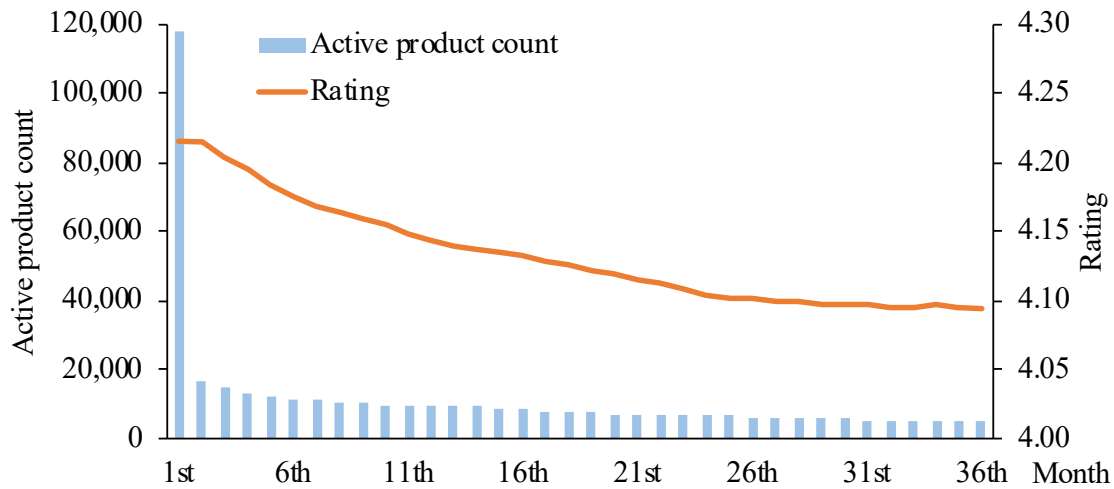


Figure 5.2: Average rating of products since launch time

This may give us a sense of how the quality of a product is generally perceived over time. The line graph in Figure 5.2 shows that the tendency is for the average rating to be the highest at launch and thereafter to fall over time. There could be various explanations, such as excitement about the product winds down, product flaws are discovered over time, etc. However, taking Figure 5.2 and Figure 5.1 together may suggest that satisfaction with earlier launched products decreases as other newly launched products appear.

Figure 5.2 also shows a histogram of products still ‘actively’ receiving ratings months from their launch. For many products, their ‘lifetime’ are rather short, as many are no longer active after one month. A cynical view is ratings decrease because inactive products are ‘better’ than long-surviving products. What we find more persuasive is that those products are inactive because they have been replaced by newer products, while others still survive as alternatives to the newer products but suffer from weaker perception.

Observations from these empirical analyses are consistent, at least not in conflict, with our hypothesis of a previously purchased product as reference. Capitalizing on these insights, we instantiate concrete formulations to build explainable recommendation models that support comparative explanations at aspect level.

$\mathcal{U}, \mathcal{P}, \mathcal{A}, \mathcal{O}$	set of all users, products, aspects, and opinions
\mathcal{L}	sentiment lexicon
\mathcal{S}	set of all purchased sequences
$S_i \in \mathcal{S}$	a purchased sequence by user i
N	the highest overall rating in the target domain
Q	user-product-aspect quality tensor
X, Q'	user-aspect attention matrix and product-aspect quality matrix
λ_x, λ_y	coefficients weigh the relative important of aspects vs. ratings
σ	logistic function
λ_d	trade-off parameter of COMPARER
α	trade-off between rating and aspect scores for ranking

Table 5.1: Main Notations in COMPARER

5.2 Notation and Formulation

The notations are summarized in Table 5.1. \mathcal{P} denotes the universal set of products of a specific category, e.g., washing machines. The set of users is denoted \mathcal{U} . A user $i \in \mathcal{U}$ assigns to a product $j \in \mathcal{P}$ a rating $r_{ij} \in \mathbb{R}_+$. Each user is also associated with S_i , which is a temporally ordered list of products rated/adopted by user i .

Let \mathcal{A} be the set of aspects, and \mathcal{O} be the set of opinion phrases. In the review accompanying a rating r_{ij} , the user may express several opinion phrases with regards to aspects. From such expressions, we extract (a, o, ρ) tuples, each representing sentiment polarity $\rho \in \{-1, +1\}$ for aspect $a \in \mathcal{A}$ with opinion phrase $o \in \mathcal{O}$. A sentence may support a tuple. Tuples across sentences within a review are to be aggregated to get user’s aspect-level sentiments (see Section 5.3). The collection of unique tuples extracted from reviews make up a contextual sentiment lexicon \mathcal{L} . To build \mathcal{L} , we leverage opinion lexicon from [26] and aspect lexicon from Microsoft Concept Graph² (see Section 5.4).

The problem can thus be stated as follows. We receive as input the set of users \mathcal{U} , products \mathcal{P} , ratings \mathcal{R} , sequences \mathcal{S} , and contextual sentiment lexicon \mathcal{L} . From these inputs, we seek a model that is capable of producing top- k personalized ranking list of products as well as an explanation associated with each recommended item. The explanation will express the tradeoff in aspects

²<https://concept.research.microsoft.com/>

between the recommended item and a reference item (previously purchased).

5.3 Methodology

We propose *Comparative Explainable Recommendation* (COMPARER). The gist is to transform observed aspect-level quality into a set of comparative constraints relating an item and previous items in the user’s adoption history. In particular, we describe two variants of this approach owing to the two modes of expressing aspect-level quality common to the explainable recommendation literature. In one mode, aspect-level quality is subjective, i.e., the perception of a product for an aspect may vary across users. In another mode, aspect-level quality is objective. It is expressed for each product.

5.3.1 Subjective Aspect-Level Quality

Aspect-Level Quality. Let Q be a tensor of dimensionality $|\mathcal{U}| \times |\mathcal{P}| \times |\mathcal{A}|$. $q_{ijk} \in Q$ represents the quality score of aspect $k \in \mathcal{A}$ that user $i \in \mathcal{U}$ assigns to product $j \in \mathcal{P}$. However, the explicit quality scores given by users are usually unavailable. Instead, they can be estimated based on sentiments extracted from textual reviews [85].

Let s_{ijk} be the aggregate (e.g., sum) of the sentiment polarity scores (the aforementioned ρ) for aspect k extracted from user i ’s review of item j . The higher it is, the more positive i assesses j on k . For instance, [85] defines a non-linear mapping from s_{ijk} to q_{ijk} .

$$q_{ijk} = \begin{cases} 0, & \text{if aspect } k \text{ is not mentioned when } i \text{ reviews } j \\ 1 + \frac{N-1}{1+e^{-s_{ijk}}}, & \text{otherwise} \end{cases} \quad (5.1)$$

where N is the highest rating score in the target domain. Realistically, Q is only partially observed. The crux of the model lies in predicting the missing values in Q .

Comparative Constraint. Let us take two products rated by user i , namely j and j' where $j \prec j'$, i.e., j is earlier in the sequence of adoption than j' . We hypothesize that for many such pairs, j' is not dominated by j . In other words, $\forall k \in \mathcal{A}, q_{ijk} \leq q_{ij'k}$ or $\exists k \in \mathcal{A}, q_{ijk} < q_{ij'k}$. Note that this is not necessarily true all the time, we study such ‘violations’ in Section 5.4. However, it holds frequently enough that we would like to impose a constraint to their corresponding predictions \hat{q}_{ijk} and $\hat{q}_{ij'k}$ (to be learnt by the prediction model) to preserve instances where $q_{ijk} < q_{ij'k}$ holds.

To this end, we favor the aspect quality comparisons where the more recently adopted product achieves superiority in some aspect. In particular, we formulate the following loss for comparative aspects, where we try to maximize the difference in the aspect quality scores between the more recent product and the earlier product.

$$L_{\text{COMPARER}_{sub}} = - \sum_{i \in \mathcal{U}} \sum_{(j \prec j') \in S_i} \sum_{\{k | q_{ijk} < q_{ij'k}\}} \ln \sigma(\hat{q}_{ij'k} - \hat{q}_{ijk}) \quad (5.2)$$

Joint Model. We seek to minimize our proposed comparative constraint loss jointly with the recommendation objective. Without loss of generality, we adopt the recommendation objective of MTER [85]. It models user-product-aspect interactions with ratings jointly as a tensor $G \in \mathbb{R}_+^{|\mathcal{U}| \times |\mathcal{P}| \times (|\mathcal{A}|+1)}$. The rating r_{ij} is appended as an additional aspect to the tensor G , i.e., $g_{ij(|\mathcal{A}|+1)} = r_{ij}$. And the aspect-level quality scores are $g_{ij(\cdot|\mathcal{A})} = q_{ij(\cdot|\mathcal{A})}$.

G is decomposed by minimizing the following loss function³:

$$L_{\text{MTER}} = \|\hat{G} - G\| - \lambda_b \sum_{i \in \mathcal{U}} \sum_{\{(i,j,l) | r_{ij} > r_{il}\}} \ln \sigma(\hat{g}_{ij(|\mathcal{A}|+1)} - \hat{g}_{il(|\mathcal{A}|+1)}) \quad (5.3)$$

The first component $\|\hat{G} - G\|$ is due to Tucker decomposition [31] on the observed elements of the tensor, where \hat{G} is the tensor reconstruction of G . The second component is due to applying

³The full objective also includes decomposition of user-aspect-opinion and item-aspect-opinion tensors. For simplicity, we only show the decomposition of user-item-aspect tensor G as the other tensors are for predicting opinion phrases. In our experiment, we use the full version of the objective function.

the Bayesian Personalized Ranking (BPR) principle [68] to the rating component of the tensor to preserve the triples (i, j, l) where we observe the rating r_{ij} to be higher than r_{il} . λ_b is a trade off parameter to balance the two types of loss. Towards joint modeling, we integrate the loss due to the comparative constraints as follows:

$$L = L_{\text{MTER}} + \lambda_d L_{\text{COMPARER}_{sub}} \quad (5.4)$$

where λ_d controls the contribution of comparative constraints.

Parameter Learning. Let Θ be the set of all learning parameters⁴. We optimize for $L_{\text{COMPARER}_{sub}}$ by minimizing the following $-\ln \sigma(\hat{q}_{ij'k} - \hat{q}_{ijk})$. The corresponding gradient is:

$$-\frac{\nabla}{\nabla \Theta} \ln \sigma(\hat{q}_{ij'k} - \hat{q}_{ijk}) \propto \frac{e^{\hat{q}_{ijk} - \hat{q}_{ij'k}}}{1 + e^{\hat{q}_{ijk} - \hat{q}_{ij'k}}} \frac{\nabla}{\nabla \Theta} (\hat{q}_{ij'k} - \hat{q}_{ijk}) \quad (5.5)$$

The complexity of enumerating comparable product pairs for each sequence is $O(|S_i|^2)$. Iterating through all aspects requires $O(|\mathcal{A}|)$. Thus, given the set of training sequences $\mathcal{S}_{\text{train}}$ with an average sequence length of \bar{S} , the overall complexity of COMPARER on a training epoch is $O(|\mathcal{S}_{\text{train}}| \cdot |\bar{S}|^2 \cdot |\mathcal{A}|)$. Nevertheless, in practice, the average sequence length⁵ and the number of aspects in each product are relatively small.

Top- k Recommendation. With the learnt parameters, we compute the ranking score for a recommended item j to user i as follows:

$$\text{RankingScore}_{ij} = \alpha \cdot \frac{\sum_{k \in C_{ij}} \hat{q}_{ijk}}{|C_{ij}|} + (1 - \alpha) \cdot \hat{r}_{ij} \quad (5.6)$$

Here, $\hat{r}_{ij} = \hat{g}_{ij(|\mathcal{A}|+1)}$ is the predicted rating for the item, which is weighted by $(1 - \alpha)$. We

⁴For simplicity of presentation, we hide all regularization from the notation. In our experiments, we use L_2 -norm for every factor as regularization.

⁵If efficiency is a concern for very long sequences, optimization strategies such as using windows or subsequences may be applicable.

also consider the effects of aspect sentiments. Let $|C_{ij}|$ be the specified number of top aspects to be considered in the prediction. Correspondingly, $C_{ij} \subseteq \mathcal{A}$ is the set of top aspects of interest by user i on item j in terms of highest \hat{q}_{ijk} . We then average these scores in C_{ij} and incorporate it with weight α . This combination is useful as we will investigate in Section 5.4.

Explanation. For each item j' in the top- k recommendation list, we provide an explanation with respect to a reference item from the user i 's previous adoption history. The choice of which item j from the user's history is to be used as reference is left as an application device. Possible heuristics could be most recent, similar, substitutable, equally-priced, etc. It could also be user-specified. The former aspect k would be one where $\hat{q}_{ijk} < \hat{q}_{ij'k}$, whereas the latter aspect k' would be one where $\hat{q}_{ijk'} > \hat{q}_{ij'k'}$ (if any). When there are more than one choice of aspect, we select randomly, though other selection criteria could also apply (e.g., highest difference).

5.3.2 Objective Aspect-Level Quality

Aspect-Level Quality. The second mode of expressing aspect-level quality is through a quality matrix $Q' \in \mathbb{R}_+^{|\mathcal{P}| \times |\mathcal{A}|}$, which is user-independent as proposed by [96]. Each element $q'_{jk} \in Q'$ may be obtained from sentiments extracted from reviews as follows:

$$q'_{jk} = \begin{cases} 0, & \text{if aspect } k \text{ is not discussed in reviews of } j \\ 1 + \frac{N-1}{1+e^{-s'_{jk}}}, & \text{otherwise} \end{cases} \quad (5.7)$$

where s'_{jk} is the aggregate (e.g., sum) of sentiment-scores of aspect k across the reviews of product j (by any reviewer).

Comparative Constraint. Let us take two products, namely j and j' where $\exists i \in \mathcal{U}, (j \prec j') \in S_i$, i.e., j' is later in the sequence of adoption than j for at least one user. We would like to impose a constraint to their corresponding predictions \hat{q}'_{jk} and $\hat{q}'_{j'k}$ (to be learnt by the prediction

model) to preserve instances where $q'_{jk} < q'_{j'k}$ holds. However, not all such constraints would be equal. Some are supported by many more sequences (users) than others. Let $c_{jj'}$ be the count of users that support the $(j \prec j')$ sequence. Intuitively, the greater $c_{jj'}$ is, the more weight it should carry in the optimization objective. Thus we apply a scaling factor of $1 + \ln(c_{jj'})$, which satisfies this objective. In particular,

$$L_{\text{COMPARER}_{obj}} = - \sum_{(j,j') \in \cup_{i \in \mathcal{U}} S_i} (1 + \ln(c_{jj'})) \sum_{\{k | q'_{jk} < q'_{j'k}\}} \ln \sigma(\hat{q}'_{j'k} - \hat{q}'_{jk}) \quad (5.8)$$

where q'_{jk} and $q'_{j'k}$ are the product aspect-level quality score of a previously bought product j and a later bought product j' .

Joint Model. To integrate the proposed comparative constraint with a compatible recommendation objective, we extend the recommendation objective of EFM [96]. In addition to product-aspect quality matrix, it models user-aspect attention matrix X by projecting the frequency t_{ik} of an aspect k mentioned by a user i .

$$x_{ik} = \begin{cases} 0, & \text{if aspect } k \text{ is not mentioned by } i \\ 1 + (N - 1) \left(\frac{2}{1 + e^{-t_{ik}}} - 1 \right), & \text{otherwise} \end{cases} \quad (5.9)$$

X and Q' are reconstructed along with ratings R by multi-matrix factorization with shared factors, minimizing the following:

$$L_{\text{EFM}} = \|UP^T - R\|^2 + \lambda_x \|\eta_1 \psi^T - X\|^2 + \lambda_y \|\eta_2 \psi^T - Q'\|^2 \quad (5.10)$$

where $U = [\eta_1 \phi_1]$ and $P = [\eta_2 \phi_2]$ are users' and products' latent factors respectively. Each is the concatenation of aspect-based factor (η_1, η_2) influenced by X, Q' and hidden factors (ϕ_1, ϕ_2) influenced by ratings. ψ are the latent factors of aspects. Coefficients λ_x and λ_y weigh the relative importance of aspects vs. ratings.

We integrate the loss functions as follows:

$$L = L_{\text{EFM}} + \lambda_d L_{\text{COMPARER}_{obj}} \quad (5.11)$$

Parameter Learning. We optimize for $L_{\text{COMPARER}_{obj}}$ where the corresponding gradient for a comparative pair (j, j') is:

$$\begin{aligned} & - (1 + \ln(c_{jj'})) \sum_{\{k|q'_{j'k} > q'_{jk}\}} \frac{\nabla}{\nabla\Theta} \ln \sigma(\hat{q}'_{j'k} - \hat{q}'_{jk}) \\ & \propto (1 + \ln(c_{jj'})) \sum_{\{k|q'_{j'k} > q'_{jk}\}} \frac{e^{\hat{q}'_{jk} - \hat{q}'_{j'k}}}{1 + e^{\hat{q}'_{jk} - \hat{q}'_{j'k}}} \frac{\nabla}{\nabla\Theta} (\hat{q}'_{j'k} - \hat{q}'_{jk}) \end{aligned} \quad (5.12)$$

Because aspect score comparisons are done at product level, instead of user level, the complexity is smaller than before: $O(|\mathcal{U}| \times |\bar{S}|^2)$ for computing the counts $c_{jj'}$ and $O(|\mathcal{P}|^2 \cdot |\mathcal{A}|)$ for parameter learning (in practice the number of compared pairs are much less than $|\mathcal{P}|^2$ due to data sparsity).

Ranking Score. The ranking score is measured as follows:

$$\text{RankingScore}_{ij} = \alpha \cdot \frac{\sum_{k \in C_i} \hat{x}_{ik} \hat{q}'_{jk}}{|C_i|N} + (1 - \alpha) \cdot \hat{r}_{ij} \quad (5.13)$$

where α is the control factor, C_i of a specified size is the set of most-cared aspects of user u_i in terms of \hat{x}_{ik} values. Other details such as top- k recommendations and explanation are similar to those described earlier in Section 5.3.1.

Dataset	#User	#Product	#Rating	#Aspect	#Opinion
Electronic	45,225	57,873	759,016	445	4,232
Toy	4,188	10,512	70,944	428	2,559
Clothing	5,200	17,895	68,262	422	1,748
Cellphone	3,216	7,807	44,492	423	2,032
Music	1,763	3,383	40,675	416	3,065

Table 5.2: Data Statistics

5.4 Experiment

As experimental objectives, we investigate whether incorporating the comparative constraints leads to improved recommendation accuracy. We also consider the resulting comparative explanations through case study and user study. Comparisons between methods are tested with one-tailed paired-sample Student’s t-test at 0.05 level. Computational efficiency is not the focus of this work. Most recommendation algorithms are learnt offline. While ranking score computation is online, its computational time is practically identical across methods being compared. Experiments were run on machine with Intel Xeon E5-2650v4 2.20 GHz CPU and 256GB RAM.

5.4.1 Setup

Datasets. For experiments, we rely on the publicly available Amazon datasets from the same source as the one used in Section 5.1 for preliminary empirical analysis. However, due to the comparative nature of the hypothesis, the modeling and learning are more appropriately conducted for distinct categories separately, as one probably does not compare a toy and a phone. Therefore, we conduct five experiments with the following categories respectively: *Electronics* (Electronic), *Toys and Games* (Toy), *Clothing* (Clothing), *Cell Phones and Accessories* (Cellphone), *Digital Music* (Music). Table 5.2 summarizes basic statistics of the datasets. For statistical sufficiency, we retain users with at least 10 ratings. Each user’s rating sequence is then split into train, validation, test with ratio 0.64 : 0.16 : 0.2 chronologically. Unknown products are excluded from validation and test sets, a uniform practice across all methods.

To extract aspects and opinions from reviews, we adopt the frequency-based approach of [18]. Using Microsoft Concepts as aspects, we retrieve top-2000 most frequently mentioned in reviews, sort them by their correlations with the ratings, and keep only top-500 (after filtering unseen aspects in validation/testing, the number comes to 400+). We select opinions associated with these aspects to construct (a, o, ρ) tuples, using the opinion lexicon from [26].

Methods and Baselines. There are two instantiations of our method, namely: COMPARER_{obj} (see Section 5.3.2) and COMPARER_{sub} (see Section 5.3.1). We note that while in Section 5.3 we describe joint models that incorporate comparative constraints with a base recommendation objective, our approach can be seen as a framework as these comparative constraints could potentially be applicable to other base recommendation objectives⁶. Therefore, the most appropriate choice of baselines would be the base recommendation objectives that we use, specifically EFM [96] for COMPARER_{obj} and MTER [85] for COMPARER_{sub} , for these would directly evaluate whether the comparative constraints produce a positive effect.

Measures. Each method produces a ranked list of recommended items. The length of each list is relative to corresponding dataset size, and is expressed as the top- $k\%$ of items in terms of the ranking score, for various $k \in \{1, 5, 10\}$. As evaluation measures, we employ multiple standard ranking metrics, such as Area Under the ROC Curve (AUC), Recall at k percentage ($\text{Recall}@k\%$), and Normalized Discount Cumulative Gain at k percentage ($\text{NDCG}@k\%$). For these metrics, a higher value indicates better performance.

Learning Details. We use grid search to find the optimal hyperparameters for the baselines EFM and MTER. For COMPARER , we then apply the same hyperparameters as the corresponding base model for parity. We fix $\lambda_x = \lambda_y = 1$ (as in the author implementation in Librec), and search for the latent dimensions $l \in \{8, 16, 32, 64, 128\}$. We further tune the coefficient λ_d in

⁶To maintain focus, we keep such explorations to future work.

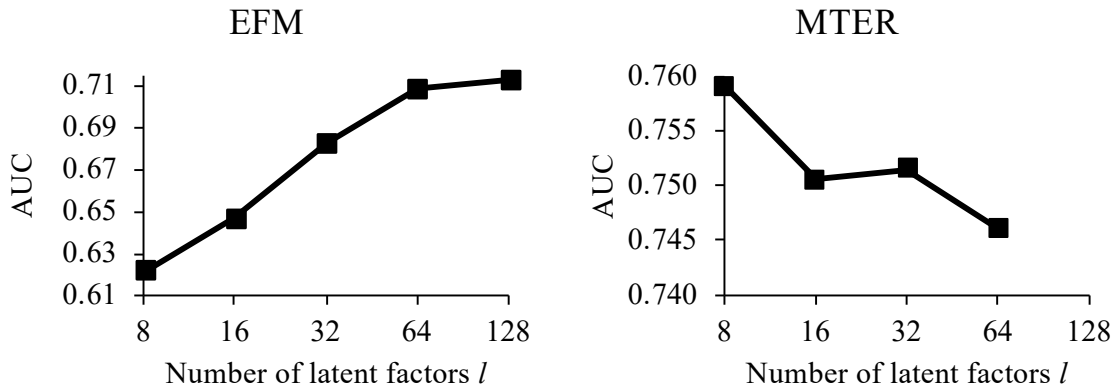


Figure 5.3: AUC performance of EFM and MTER while varying number of latent factors l on Electronic data

the candidate set of $\{0.01, 0.1, 1, 10, 100\}$. For each method, the setting with the best AUC on validation set is selected.

Figure 5.3 illustrates the performance of the base models while varying the latent dimensionality l in terms of AUC on the largest Electronic data (we observe similar trends on other datasets as well). EFM achieves better AUC with greater dimensionality l . The opposite is true for MTER, yet it requires much more time for training. So we set $l = 128$ for EFM and $l = 8$ for MTER as default, which we apply to our methods as well. To speed up training, we load pretrained weights from the respective base model and continue training with the added constraints. For parity purpose, we further verify that as the base models have indeed converged, further continuing their training does not add any value.

5.4.2 Ranking Performance

First, we investigate whether adding the comparative constraints improve the ranking performance of the base models. Table 5.3 shows the results for COMPARER_{obj} and its baseline EFM. We observe that on all metrics, across all datasets, COMPARER_{obj} improves upon the ranking performance of EFM consistently and in a statistically significant manner. We attribute this to the contribution of the comparative constraints based on historical reference. In turn, Table 5.4 shows the ranking performance of COMPARER_{sub} and its baseline MTER. It substantially echoes

Dataset	Model	AUC	Recall@ $k\%$			NDCG@ $k\%$		
			1	5	10	1	5	10
Electronic	EFM	0.717	0.107	0.277	0.393	0.022	0.044	0.057
	COMPARER _{obj}	0.759 [§]	0.176 [§]	0.362 [§]	0.474 [§]	0.038 [§]	0.062 [§]	0.074 [§]
Toy	EFM	0.580	0.030	0.106	0.189	0.009	0.021	0.033
	COMPARER _{obj}	0.656 [§]	0.042 [§]	0.157 [§]	0.268 [§]	0.014 [§]	0.033 [§]	0.049 [§]
Clothing	EFM	0.579	0.036	0.114	0.189	0.009	0.020	0.028
	COMPARER _{obj}	0.611 [§]	0.059 [§]	0.154 [§]	0.233 [§]	0.016 [§]	0.030 [§]	0.039 [§]
Cellphone	EFM	0.652	0.045	0.162	0.266	0.012	0.032	0.046
	COMPARER _{obj}	0.701 [§]	0.069 [§]	0.214 [§]	0.334 [§]	0.022 [§]	0.046 [§]	0.062 [§]
Music	EFM	0.641	0.040	0.158	0.257	0.019	0.046	0.064
	COMPARER _{obj}	0.678 [§]	0.060 [§]	0.200 [§]	0.320 [§]	0.029 [§]	0.061 [§]	0.083 [§]

[§] denotes statistically significant improvements. Highest values are in **bold**

Table 5.3: Performance of EFM and COMPARER_{obj}

the observations above that supports the outperformance of COMPARER_{sub} over its baseline. Much of the outperformance are also statistically significant, save for a couple of pockets (e.g., Recall@1% on the smaller datasets Clothing and Music) where the differences still exist but in a smaller way.

5.4.3 Comparative Constraints

Constraint Violation. Earlier, we motivate the comparative constraints with the hypothesis whereby it is unlikely that an item j' that a user rates later is ‘dominated’ by another item j rated earlier. Seeking some measure of validation, we now analyze the number of occurrences in which this hypothesis is violated. To define violation, we use the ground-truth matrix Q' for objective aspect-level quality. For subjective aspect-level quality, a matrix is obtained from flattening Q by averaging the values across users.

Table 5.5 shows the total number pairs involving two products, one of which is rated later than the other by a user. Suppose that $V(\cdot)$ is a counting function that takes in all these pairs and the aspect quality weights as input, and reports the number of violating pairs. We express these numbers both as absolute count as well as a percentage of the total number of pairs. Interestingly,

Dataset	Model	AUC	Recall@ $k\%$			NDCG@ $k\%$		
			1	5	10	1	5	10
Electronic	MTER	0.759	0.157	0.337	0.448	0.035	0.058	0.070
	COMPARER _{sub}	0.797 [§]	0.185 [§]	0.398 [§]	0.520 [§]	0.041 [§]	0.069 [§]	0.083 [§]
Toy	MTER	0.727	0.066	0.217	0.359	0.020	0.044	0.064
	COMPARER _{sub}	0.747 [§]	0.093 [§]	0.278 [§]	0.422 [§]	0.029 [§]	0.059 [§]	0.079 [§]
Clothing	MTER	0.671	0.069	0.189	0.287	0.017	0.034	0.045
	COMPARER _{sub}	0.680 [§]	0.071	0.200 [§]	0.297 [§]	0.017	0.035 [§]	0.046 [§]
Cellphone	MTER	0.757	0.113	0.296	0.425	0.036	0.066	0.084
	COMPARER _{sub}	0.787 [§]	0.129 [§]	0.337 [§]	0.474 [§]	0.043 [§]	0.077 [§]	0.095 [§]
Music	MTER	0.844	0.128	0.380	0.548	0.057	0.114	0.146
	COMPARER _{sub}	0.848 [§]	0.130	0.391 [§]	0.564 [§]	0.059	0.119 [§]	0.151 [§]

[§] denotes statistically significant improvements. Highest values are in **bold**

Table 5.4: Performance of MTER and COMPARER_{sub}

Dataset	#Pairs	$V(Q')$	$\frac{V(Q')}{\#Pairs}$	$V(Q)$	$\frac{V(Q)}{\#Pairs}$
Electronic	4,692,596	57,668	1.23%	30,867	0.66%
Toy	479,786	11,602	2.42%	10,476	2.18%
Clothing	258,357	14,081	5.45%	13,226	5.11%
Cellphone	219,851	8,024	3.65%	5,751	2.61%
Music	509,302	6,515	1.28%	2,481	0.49%
Total	6,159,892	97,890	1.59%	62,801	1.02%

Table 5.5: Constraint violations analysis. Counting function $V(\cdot)$ takes all pairs and the aspect quality weights as input and reports number of pairs violating the constraint

the stated hypothesis seems to hold for the vast majority of pairs. The violations amount to a single-digit percentage value, which across all datasets come to less than 2% of all pairs.

Effect of Constraint Coefficient λ_d . We incorporate COMPARER constraints with a coefficient weight λ_d to learn model parameters that would preserve the constraint for as many pairs as possible. Table 5.6 tabulates the ranking performance and violation count of COMPARER_{obj} at various values of λ_d . When λ_d is zero, we are optimizing only for the recommendation objective. Interestingly, as we increase λ_d , the number of violations (last column) generally decreases, which means the imposed constraints are taking effect. The ranking performance also initially improves, though with too high λ_d it may hurt ranking performance as it downweights the recommendation objective. Table 5.7 presents the results for COMPARER_{sub} with largely the same

Dataset	λ_d	AUC	Recall@k%			NDCG@k%			$V(\hat{Q}')$
			1	5	10	1	5	10	
Electronic	0.01	0.755	0.182	0.357	0.463	0.042	0.065	0.077	91,669
	0.1	0.759	0.176	0.362	0.474	0.038	0.062	0.074	75,528
	1	0.749	0.151	0.348	0.459	0.029	0.055	0.067	45,982
	10	0.735	0.103	0.296	0.412	0.021	0.046	0.059	48,401
Toy	0.01	0.647	0.047	0.160	0.259	0.015	0.033	0.047	10,772
	0.1	0.656	0.042	0.157	0.268	0.014	0.033	0.049	10,111
	1	0.649	0.033	0.156	0.269	0.011	0.032	0.047	9,919
	10	0.624	0.036	0.139	0.233	0.013	0.030	0.043	9,474
Clothing	0.01	0.606	0.053	0.151	0.229	0.015	0.028	0.038	14,053
	0.1	0.611	0.059	0.154	0.233	0.016	0.030	0.039	12,043
	1	0.612	0.054	0.153	0.237	0.015	0.028	0.038	12,294
	10	0.605	0.051	0.150	0.237	0.012	0.026	0.036	11,505
Cellphone	0.01	0.697	0.072	0.218	0.331	0.023	0.047	0.062	7,698
	0.1	0.701	0.069	0.214	0.334	0.022	0.046	0.062	6,739
	1	0.698	0.053	0.202	0.320	0.017	0.041	0.057	6,442
	10	0.689	0.044	0.169	0.294	0.014	0.034	0.051	5,598
Music	0.01	0.672	0.056	0.195	0.309	0.025	0.057	0.078	5,634
	0.1	0.678	0.060	0.200	0.320	0.029	0.061	0.083	5,074
	1	0.675	0.044	0.179	0.308	0.023	0.054	0.078	3,948
	10	0.648	0.036	0.142	0.248	0.016	0.041	0.060	3,875

Better values are in **bold**

Table 5.6: Effect of Constraint Coefficient λ_d on COMPARER_{obj}

conclusion as well.

To see how COMPARER retains the original violations as in the base models, not only in terms of the violation counts, but also whether it is identifying the ‘correct’ violations, Table 5.8 and Table 5.9 show that COMPARER achieves lower number of constraint violations than the baselines that do not optimize for this directly.

To further clarify the degree of agreement between the estimated scores obtained after training (\hat{Q}' or \hat{Q}) and the ground-truth scores in training data (Q' or Q), we evaluate the Recall = $\frac{V(Q') \cap V(\hat{Q}')}{V(\hat{Q}')}$, Precision = $\frac{V(Q') \cap V(\hat{Q}')}{V(Q')}$, F-Measure (similar formula applies to Q).

Given the large number constraint violations by the base models, perhaps it is not surprising that they have higher recall. However, much of the recovered violations may not be correct, as reflected by their lower precision as compared to COMPARER . When we take the recall and precision together, their harmonic mean or F-measure shows that COMPARER performs better

Dataset	λ_d	AUC	Recall@ $k\%$			NDCG@ $k\%$			$V(\hat{Q})$
			1	5	10	1	5	10	
Electronic	0.1	0.759	0.150	0.331	0.445	0.033	0.056	0.069	2,029,774
	1	0.774	0.163	0.356	0.476	0.036	0.061	0.074	2,138,013
	10	0.794	0.180	0.388	0.512	0.040	0.067	0.081	1,062,211
	100	0.797	0.185	0.398	0.520	0.041	0.069	0.083	814,568
Toy	0.1	0.725	0.065	0.214	0.357	0.019	0.043	0.063	246,307
	1	0.733	0.072	0.232	0.373	0.022	0.048	0.068	229,863
	10	0.747	0.093	0.278	0.422	0.029	0.059	0.079	106,607
	100	0.747	0.082	0.263	0.410	0.024	0.054	0.074	70,228
Clothing	0.1	0.666	0.069	0.187	0.287	0.017	0.034	0.046	131,862
	1	0.670	0.069	0.192	0.293	0.017	0.034	0.046	130,323
	10	0.680	0.071	0.200	0.297	0.017	0.035	0.046	82,484
	100	0.678	0.065	0.190	0.297	0.016	0.033	0.046	61,998
Cellphone	0.1	0.751	0.112	0.282	0.409	0.036	0.064	0.081	117,560
	1	0.762	0.129	0.313	0.432	0.042	0.072	0.089	109,877
	10	0.783	0.156	0.359	0.477	0.053	0.086	0.102	53,091
	100	0.787	0.129	0.337	0.474	0.043	0.077	0.095	22,119
Music	0.1	0.840	0.127	0.381	0.554	0.059	0.117	0.149	257,273
	1	0.844	0.129	0.386	0.560	0.060	0.118	0.151	249,079
	10	0.848	0.130	0.391	0.564	0.059	0.119	0.151	101,376
	100	0.833	0.126	0.367	0.531	0.058	0.113	0.144	124,574

Better values are in **bold**

Table 5.7: Effects of Constraint Coefficient λ_d on COMPARER_{sub}

in recovering the violations.

5.4.4 Incorporating Aspects in Ranking Scores

To verify that the aspects do participate meaningfully in the recommendation, we tune different values α in the range of $[0, 1]$ with step size 0.1 and the number of top aspects in a candidate set of $\{10, 20, 30, 50, 100, 200, 300, 400, \text{all aspects}\}$. Table 5.10 shows the setting with the best performance on various datasets for COMPARER_{sub} and COMPARER_{obj} . Evidently, for all datasets, we have $\alpha > 0$, which means that aspects are indeed helpful. The number of aspects to take into account in the prediction is dataset- and method-dependent.

Dataset	Model	$V(\hat{Q}')$	Recall	Precision	F-Measure
Electronic	EFM	92,491	0.810	0.505	0.622
	COMPARER _{obj}	75,528	0.875	0.668	0.758
Toy	EFM	17,799	0.888	0.579	0.701
	COMPARER _{obj}	10,111	0.857	0.983	0.916
Clothing	EFM	21,718	0.870	0.564	0.684
	COMPARER _{obj}	12,294	0.872	0.999	0.931
Cellphone	EFM	11,523	0.878	0.611	0.721
	COMPARER _{obj}	6,739	0.825	0.982	0.897
Music	EFM	10,090	0.838	0.541	0.658
	COMPARER _{obj}	5,074	0.750	0.962	0.843

Better values are in **bold**

Table 5.8: Constraint Violations: EFM vs. COMPARER_{obj}

Dataset	Model	$V(\hat{Q})$	Recall	Precision	F-Measure
Electronic	MTER	2,033,981	0.976	0.015	0.029
	COMPARER _{sub}	814,568	0.867	0.033	0.063
Toy	MTER	229,456	0.971	0.044	0.085
	COMPARER _{sub}	106,607	0.892	0.088	0.160
Clothing	MTER	123,797	0.944	0.101	0.182
	COMPARER _{sub}	82,484	0.877	0.141	0.242
Cellphone	MTER	99,014	0.946	0.055	0.104
	COMPARER _{sub}	22,119	0.648	0.168	0.267
Music	MTER	198,437	0.800	0.010	0.020
	COMPARER _{sub}	101,376	0.541	0.013	0.026

Better values are in **bold**

Table 5.9: Constraint Violations: MTER vs. COMPARER_{sub}

Dataset	COMPARER _{obj}		COMPARER _{sub}	
	α	#Top Aspects	α	#Top Aspects
Electronic	0.8	300	0.4	200
Toy	0.7	20	0.5	all aspects
Clothing	0.8	10	0.3	10
Cellphone	0.6	10	0.3	300
Music	0.7	20	0.3	400

Table 5.10: Aspects in Ranking Score: α and Number of Top Aspects

User: A15N56ZCTHRB73

Recommended product: B00F6AVFK8
The Oontz XL - Cambridge SoundWorks Most Powerful Portable, Wireless, Bluetooth Speaker



Previously bought product: B00AI5V3CQ
The Oontz Angle Ultra Portable Wireless Bluetooth Speaker - Better Sound, Better Volume, Incredible Online Price - The Perfect Speaker to take everywhere with you this summer (Blue)



Explanation:

EFM: You might be interested in **sound**, on which this product performs well.

You might be interested in **purpose**, on which this product performs poorly.

ComparER_{obj}: Product B00F6AVFK8 is better at **quality** than B00AI5V3CQ. But worse at **sound**.

Figure 5.4: Example Explanations by EFM and COMPARER_{obj}

5.5 Comparative Explanation

One of the objectives is to explain a recommended item by way of comparison to a reference item (another item previously rated or purchased by the user). In this section, we show a couple of examples of such explanations and discuss a user study.

5.5.1 Case Study

For the first example in Figure 5.4, we recommend a bluetooth speaker *Oontz XL* to the user. The explanation generated by the baseline EFM for this product is evaluative, speaking of the aspects *sound* which is positive and *purpose* which is negative. It offers no hint as for how this product may compare to any other. A comparative explanation relies on a reference item, which we propose to be a previously rated product. In this particular case, one of the previously rated products in the category was another bluetooth speaker *Oontz Angle*, which is a smaller

User: ACO3U8DT64IV6

Recommended product: B00GN6QZ0Y
Mpow 3.1Amps 15.5W Dual Port Backlight
USB Car Charger for iPhone 5s 5c 5 4s 4 iPad
1 2 3 5 Air Mini Samsung Galaxy S4 S3 S2
Galaxy Note 3 2 HTC One X V S and More
(White and Blue)



Previously bought product: B000S5Q9CA
Motorola Vehicle Power Adapter micro-USB
Rapid Rate Charger



Explanation:

MTER: Its **phone** is mistakenly. Its **case** is mistakenly.

COMPARER_{sub}: Product B00GN6QZ0Y is better at **design** than B000S5Q9CA. But worse at **quality**.

Figure 5.5: Example Explanations by MTER and COMPARER_{sub}

model than the recommended item. Using our approach COMPARER_{obj}, we identify *quality* as an aspect for which the recommended item is better, and *sound* as an aspect for which it is worse than the reference item. Since the user would have been familiar with the reference item (previously rated), this may offer more information than a standalone explanation.

For a second example involving COMPARER_{sub} and its baseline MTER, Figure 5.5 shows the case of a user being recommended a car charger of *Mpow* brand. MTER's explanation is based on opinion phrases. In this case, it identifies two pertinent aspects: *phone* and *case* and the opinion phrase *mistakenly*. In contrast, our approach is to present a reference item, which is a previously purchased car charger of *Motorola* brand. The explanation by COMPARER_{sub} alludes to the recommended item being better at *design* (it is more compact and cableless) but worse at *quality* (it is of a less well-known brand than the reference item).

+ reference product	Method	Score	Method	Score
No	EFM (original)	2.12	MTER (original)	2.06
Yes	EFM (enhanced)	2.24	MTER (enhanced)	2.05
Yes	COMPARER _{obj}	3.29 [§]	COMPARER _{sub}	3.06 [§]

[§] p -value < 0.01. Highest value are in **bold**

Table 5.11: Analysis of User Study

5.5.2 User Study

We conduct a user study with 25 examples (5 product recommendations from each category). Since the focus in this section is on the explanation, rather than the relative accuracy of methods, we consider recommended items from users’ test data, with reference items from their training data. We generate explanation for the recommended product by COMPARER and its base model for every example. As seen in the case studies, the original versions of EFM and MTER generate explanation for only the recommended product. To give them the benefit of comparison, we further include enhanced versions of these baselines by showing explanations for the reference products as well, as in our approach.

We then conduct independent surveys, each containing 25 examples of different recommendation explanations generated from different models (selected randomly and presented blindly). The study was completed by 20 annotators, who are neither the authors nor having any knowledge of the objective of the study. We ask every annotator to rate their opinion on the generated explanation with the following question (adopted from [85]):

Does the explanation help you know more about the recommended product?

Each explanation is seen by at least 3 different people. A participant chooses from five-point Likert scale, from 1 (strongly disagree) to 5 (strongly agree). The average scores are reported in Table 5.11. Both COMPARER variants received significantly better scores than the original and the enhanced versions of the base methods.

While we are aware of general limitations of user studies, we presume that similar limitations apply to both our approach and the baselines. The consistency in which users find favor with

the proposed explanations provide some evidence for the promising nature of COMPARER at providing comparative explanations.

5.6 Summary

In this chapter, we approach explainable recommendation from the perspective where an explanation compares the recommended item with a reference item (previously adopted product). The proposed COMPARER incorporates comparative constraints into explainable recommendation models. Experiments on datasets of five categories show that COMPARER enhances the performance of ranking prediction.

Chapter 6

Selecting Comparative Sets of Reviews

Across Multiple Items

E-commerce is now the predominant means for procuring items. Because e-commerce sites are not severely limited by inventory shelf spaces, unlike brick-and-mortar stores, they can offer many options for every consumer intent. Given the large number of alternatives to consider, and little prior experience with them, consumers resort to product reviews to glean as much information as they can from the experiences of others as documented in the reviews.

Reviews are so much a part of the e-commerce landscape now that virtually every store features reviews. So much so, that nowadays it is common to find products with thousands of reviews, if not more. What was originally a mechanism to address the paradox of choice (of which products to purchase) has now itself turned into another paradox of choice (of which reviews to read).

If a consumer only has time to read a few reviews, which among the many (potentially thousands) should they read? In the literature, this question has been addressed from multiple angles (see Section 2). One option is to let users vote on which reviews have been helpful, but this may not give a fair chance to all reviews as even the voters may have only seen a few reviews. Another option is to create a summary of all the reviews, but this summary, either being crafted by a

machine learning model or assembled from many reviews, may not have the original authenticity of a genuine review.

Review Selection. In this work, we follow the line of research in selecting a small number k of reviews that are “representative” of the full set of reviews of a given product. There are various ways to define representativeness as surveyed in Section 2. One that is particularly relevant is *characteristic* review selection [37], which seeks to find a subset of reviews that collectively cover both positive and negative opinions of product aspects in a proportion that is close to the overall. Intuitively, by reading the few selected reviews, a consumer would be well-versed in considering the trade-offs associated with a product. Notably, in the existing literature, review selection is conducted for an individual product independently.

Comparative Review Selection. We posit that a consumer’s decision making is not simply binary in the sense of whether to purchase a product. Rather, it is usually comparative in the sense of which among a few alternatives to decide upon. For instance, on certain e-commerce sites such as Amazon.com, when consumers are viewing a target item (e.g., Canon EOS Rebel T7 DSLR Camera¹), they may be presented with a number of “similar” items, ostensibly due to similarity in attribute or specifications, as illustrated in Figure 6.1. There are yet other means of identifying comparative items such as also bought items, also viewed items, etc.

As shown in Figure 6.1, each item could have hundreds to thousands of reviews. Beyond the hard specs, consumers would likely still wish to read the reviews. Given a target product and a number of comparative products, our primary focus in this work is on selecting reviews from the given products in such a way that the selected reviews would be representative of the respective products, and simultaneously *the selected reviews would cover similar aspects that would facilitate comparison across those products*. This latter objective is novel to this work. It also gives rise to a new problem formulation as what used to be a combinatorial selection across reviews

¹<https://www.amazon.com/Canon-Rebel-T7-18-55mm-II/dp/B07C2Z21X5>

Compare with similar items





				
<p>This Item Canon EOS Rebel T7 DSLR Camera with 18-55mm Lens Built-in Wi-Fi 24.1 MP CMOS Sensor DIGIC 4+ Image Processor and Full HD Videos</p> <p>#1 Best Seller</p>	<p>Canon EOS Rebel T7 DSLR Camera Bundle with Canon EF-S 18-55mm f/3.5-5.6 II Lens + 2pc SanDisk 32GB Memory Cards + Accessory Kit</p>	<p>Canon EOS Rebel T7 DSLR Camera 2 Lens Kit with EF18-55mm + EF 75-300mm Lens, Black</p>	<p>Canon EOS Rebel T8i EF-S 18-55mm IS STM Lens Kit, Black</p>	
<p>Add to Cart</p>	<p>Add to Cart</p>	<p>Add to Cart</p>	<p>Add to Cart</p>	
<p>Customer Rating</p>	<p>★★★★☆ (2294)</p>	<p>★★★★☆ (3004)</p>	<p>★★★★☆ (1170)</p>	<p>★★★★☆ (573)</p>
<p>Price</p>	<p>\$479⁰⁰</p>	<p>\$579⁰⁰</p>	<p>\$549⁸⁹</p>	<p>\$899⁰⁰</p>
<p>Sold By</p>	<p>Focus Camera LLC</p>	<p>PAGING ZONE</p>	<p>Southtown Camera</p>	<p>AAAA Universe</p>
<p>Color</p>	<p>Black</p>	<p>Black</p>	<p>Black</p>	<p>Black</p>
<p>Continuous Shooting Speed</p>	<p>3 frames_per_second</p>	<p>3.00</p>	<p>3 frames_per_second</p>	<p>7 frames_per_second</p>
<p>Screen Size</p>	<p>3 inches</p>	<p>3.0 inches</p>	<p>3 inches</p>	<p>3 inches</p>
<p>Focus Type</p>	<p>Auto Focus</p>	<p>manual-and-auto</p>	<p>Auto Focus</p>	<p>Auto Focus</p>
<p>Image Stabilization</p>	<p>true</p>	<p>—</p>	<p>true</p>	<p>—</p>
<p>ISO Range</p>	<p>100-6400</p>	<p>100-12800</p>	<p>100-6400</p>	<p>Auto, 100-25600</p>
<p>Item Dimensions</p>	<p>3.1 x 5.1 x 4 inches</p>	<p>5.09 x 3.99 x 3.06 inches</p>	<p>3.1 x 5.1 x 4 inches</p>	<p>3 x 5.16 x 4.04 inches</p>
<p>Item Weight</p>	<p>1.04 lbs</p>	<p>—</p>	<p>3.00 lbs</p>	<p>2.90 lbs</p>
<p>Max Resolution</p>	<p>24.1 megapixels</p>	<p>24.1 megapixels</p>	<p>24.1 megapixels</p>	<p>24.1 megapixels</p>
<p>Optical Sensor Resolution</p>	<p>24.1 megapixels</p>	<p>24.1 megapixels</p>	<p>24.1 megapixels</p>	<p>24.1 megapixels</p>
<p>Optical Zoom</p>	<p>0x</p>	<p>3.00x</p>	<p>0x</p>	<p>0x</p>
<p>Photo Sensor Size</p>	<p>APS-H</p>	<p>APS-C</p>	<p>APS-H</p>	<p>APS-C</p>
<p>Style</p>	<p>18-55mm</p>	<p>—</p>	<p>*Special 2 Lens Kit Deal*</p>	<p>—</p>
<p>Video Capture Resolution</p>	<p>1080p</p>	<p>1080p</p>	<p>1080p</p>	<p>2160p</p>
<p>Viewfinder Type</p>	<p>Optical</p>	<p>Optical</p>	<p>Optical</p>	<p>Optical</p>
<p>Wireless Communication Technology</p>	<p>BuiltIn; 802.11b/g/n with NFC</p>	<p>Wi-Fi</p>	<p>BuiltIn; 802.11b/g/n with NFC</p>	<p>Bluetooth, Wi-Fi</p>

Figure 6.1: “Compare with similar items” on Amazon.com

of one product now becomes combinatorial explosion across multiple products. We formulate synchronized review selection objectives and propose algorithms towards approximating them.

While a consumer is then presented only with a small number of reviews, reading a few reviews across multiple products could still be taxing on the mind. Thus, to further ease the cognitive load on consumers, as a secondary objective, we would build on the aforementioned review selection objective to narrow down the given (long) list of comparative products to a smaller sized list of core comparative products. By formulating the products or items as a graph of vertices, with the edge weights reflecting the similarities across their selected reviews, we turn the problem into finding top- k items with the heaviest weight including the target item.

Contribution. We make several contributions in this work. First, we propose a novel review selection problem for selecting review sets for multiple products simultaneously. Second, we formulate the objective function that synchronizes the review selection process and design efficient algorithms to solve this objective function. Third, we describe an efficient heuristic approximation to find top- k similar items among the candidate comparative items. Fourth, we conduct experiments on real world data to validate the efficacies of the proposed algorithm against comparable baselines.

6.1 Preliminaries

Table 6.1 lists the main notations used in this chapter. Let $\mathcal{P} = \{p_1, p_2, \dots, p_n\}$ be the collection of n items. Each item p_i has a collection of reviews \mathcal{R}_i discussing aspects from a universal set of z aspects $\mathcal{A} = \{a_1, a_2, \dots, a_z\}$. A typical review only comments on a subset of these aspects, expressing positive or negative opinion. Given a collection of reviews $\mathcal{S}_i \subseteq \mathcal{R}_i$, we use $\pi(\mathcal{S}_i)$ to denote the opinion vector that represents the distribution of opinions of \mathcal{S}_i . Each aspect has two possible opinions: positive and negative. Thus, $\pi(\mathcal{S}_i)$ has dimension of $2 \times z$.

In a nutshell, our work generalizes the CHARACTERISTIC REVIEW SELECTION [37] that is

\mathcal{P}	set of n products $\{p_1, p_2, \dots, p_n\}$
\mathcal{R}_i	set of all reviews of item p_i
$\mathcal{S}_i \subseteq \mathcal{R}_i$	a subset reviews of \mathcal{R}_i
m	maximum number of reviews to be selected
k	top- k most similar items to be selected
$\pi(\mathcal{S}_i)$	opinion distribution vector of \mathcal{S}_i
$\phi(\mathcal{S}_i)$	aspect distribution vector of \mathcal{S}_i
τ_i	target opinion distribution vector for item p_i
Γ	target aspect distribution vector
$\Delta(x, y)$	distance of two vector x and y , i.e., L2 distance
λ	control factor of opinion over aspect
μ	control factor of comparisons among items

Table 6.1: Main Notations

designed for a single item.

Problem 1. CHARACTERISTIC-REVIEW SELECTION (CRS). Given a collection of reviews \mathcal{R}_i , a target vector τ_i and an integer number m , find $\mathcal{S}_i \subseteq \mathcal{R}_i$ such that $|\mathcal{S}_i| \leq m$ and

$$\Delta(\tau_i, \pi(\mathcal{S}_i)) \quad (6.1)$$

is minimized.

Where m is the maximum number of reviews to be selected, τ_i is the target opinion distribution vector that we would like the selected subset of review \mathcal{S}_i best representing, and Δ is a distance function between two vectors. Specifically, for two vectors x and y (l -dimensional), we can compute their distance using the L_2^2 norm of their difference:

$$\Delta(x, y) = L_2^2(x - y) = (x - y)^2 = \sum_{i=1}^l (x_i - y_i)^2 \quad (6.2)$$

The CRS problem was shown to be NP-complete [37]. The CRS formulation focuses on only one item. When there are two or more items, users seek information to compare among these items for consideration. In the following section, we address the problem of selecting comparative sets of reviews for multiple items simultaneously, optimizing for similarity among the selected sets.

6.2 Comparative Review Sets Selection

We first present our problem formulations, then describe the proposed algorithm to approximate the otherwise intractable problem.

6.2.1 Problem Formulations

Let $\phi(\mathcal{S}_i)$ be the vector representing aspect distribution of \mathcal{S}_i (just the aspects, irrespective the opinion of individual items). When comparing two items, we often base on common aspects of both items regardless of their opinions to see how they are different from each other. For every item, we would like to select a subset of reviews that characterize the item well. In addition, we also want the selected sets of reviews to be similar to one another, e.g., discussing same aspects, so we can compare more directly.

(COMPARESETS) Specifically, for any two items p_i and p_j , the selected sets are \mathcal{S}_i and \mathcal{S}_j respectively, we would like to minimize $\Delta(\phi(\mathcal{S}_i), \phi(\mathcal{S}_j))$, minimizing the distance between two aspect distribution vectors of \mathcal{S}_i and \mathcal{S}_j . We also use the notion of a target aspect vector Γ , acting as an independent optimization goal, i.e., aspect vector of the target item.

The formal problem formulation is as follows:

Problem 2. COMPARATIVE REVIEW SETS SELECTION (COMPARESETS). We are given a collection of n products $\mathcal{P} = \{p_1, p_2, \dots, p_n\}$, where p_1 is regarded as the target product and p_2 to p_n as comparative products. Every product p_i has a collection of reviews \mathcal{R}_i , a target opinion vector τ_i . For a target aspect vector Γ and an integer number m , find $\mathcal{S}_i \subseteq \mathcal{R}_i$ for every product p_i such that $|\mathcal{S}_i| \leq m$ and

$$\sum_{i=1}^n \Delta(\tau_i, \pi(\mathcal{S}_i)) + \lambda^2 \sum_{i=1}^n \Delta(\Gamma, \phi(\mathcal{S}_i)) \quad (6.3)$$

is minimized.

Where $\lambda \geq 0$ is the tradeoff factor between distance of opinion vectors and distance of aspect vectors. Equation 6.3 can be solved separately for each item by minimizing:

$$\Delta(\tau_i, \pi(\mathcal{S}_i)) + \lambda^2 \Delta(\Gamma, \phi(\mathcal{S}_i)) \quad (6.4)$$

Based on the distance metric in Equation 6.2, we can rewrite Equation 6.4 as follows:

$$\Delta([\tau_i; \lambda \cdot \Gamma], [\pi(\mathcal{S}_i); \lambda \cdot \phi(\mathcal{S}_i)]) \quad (6.5)$$

Where $[\tau_i; \lambda \cdot \Gamma]$ is the concatenation between τ_i and $\lambda \cdot \Gamma$. Given the analogues of Equation 6.1 and Equation 6.5, since CRS is NP-complete, the proposed COMPARESETS problem is also NP-complete.

(COMPARESETS+) The above formulation in Equation 6.3 relates the comparative items through their respective commonality in aspects with the target item. This could inadvertently results in a situation where different comparative items cover different aspects of the target item. To address this, we need to incorporate the direct commonality between any pair of comparative items. We thus further extend the COMPARESETS objective as follows:

Problem 3. SYNCHRONIZED COMPARATIVE REVIEW SETS SELECTION (COMPARESETS+).

We are given a collection of n products $\mathcal{P} = \{p_1, p_2, \dots, p_n\}$, where p_1 is regarded as the target product and p_2 to p_n as comparative products. Every product p_i has a collection of reviews \mathcal{R}_i , a target opinion vector τ_i . For a target aspect vector Γ and an integer number m , find $\mathcal{S}_i \subseteq \mathcal{R}_i$ for every product p_i such that $|\mathcal{S}_i| \leq m$ and

$$\sum_{i=1}^n \Delta(\tau_i, \pi(\mathcal{S}_i)) + \lambda^2 \sum_{i=1}^n \Delta(\Gamma, \phi(\mathcal{S}_i)) + \mu^2 \sum_{i=1}^{n-1} \sum_{j=i+1}^n \Delta(\phi(\mathcal{S}_i), \phi(\mathcal{S}_j)) \quad (6.6)$$

is minimized.

Where $\mu \geq 0$ is the control factor of comparisons among items.

We can reduce COMPARESETS to COMPARESETS+ by limiting the number of items to only a single item. Solving COMPARESETS+ is at least as difficult as COMPARESETS, thus it is also NP-complete.

6.2.2 Integer-Regression Algorithm

In this section, we design heuristic algorithms to approximate the proposed formulations COMPARESETS and COMPARESETS+.

Let s be a $|\mathcal{R}_i|$ -dimensional vector, each element is $s_j \in \{0, 1\}$, where $s_j = 1$ indicates the review $r_j \in \mathcal{R}_i$ being selected into the solution set \mathcal{S}_i . For an item p_i , let W be an $(m+z) \times |\mathcal{R}_i|$ matrix, in which each entry $W_{ij} = 1$ iff opinion o_i appears in review r_j and $W_{(m+i')j} = \lambda$ iff aspect $a_{i'}$ appears in review r_j .

In practice, W may contain duplicate columns; these are irrelevant for regression, so we form \tilde{W} with distinct columns. These duplication columns in W correspond to distinct reviews that may be helpful in approximating $[\tau_i, \Gamma]$; hence we keep track of the number of such duplicate columns by remembering for each column i of \tilde{W} its multiplicity c_i in W . For a visual illustration, refer to Figure 6.2.

A common strategy to solve this is first solving the continuous version of the optimization problem, then transforming the continuous solution into the closest discrete one. This is a well known strategy that has been shown to be effective for combinatorial optimization problems and has been applied to solve CRS problem [37]. Analogously, we apply this algorithm to solve Equation 6.4.

This regression algorithm works in two steps, which are repeated for all values from 1 to m :

For $\ell = 1$ to m

Step 1: Form a nonnegative real-valued vector x such that $\Delta([\tau_i; \Gamma], Wx)$ is small, and the number of nonzero elements of x is not larger than ℓ .

Step 2: Form a nonnegative integer-valued vector s representing k reviews that together approxi-

mate \mathbf{x} in distribution. That is, find \tilde{s} such that $\forall i, \tilde{s}_i \leq c_i, \|\mathbf{s}_i\|_1 \leq m$, and $\|\frac{\tilde{s}}{\|\tilde{s}\|_1} - \frac{\mathbf{x}}{\|\mathbf{x}\|_1}\|_1$ is minimized.

Similarly to [37], we adopt NONNEGATIVE ORTHOGONAL MATCHING PURSUIT (NOMP) algorithm for **Step 1**. In general, we can use other algorithms to solve this regression problem.

In **Step 2**, we construct the closest possible discrete approximation to the output of **Step 1**. The problem can be expressed as follows: Let the number of nonzero elements in \mathbf{x} be q . Given the nonnegative real-valued vector $\mathbf{v} \in \mathbb{R}^q$ having $\|\mathbf{v}\|_1 = 1$, and a set of integers $\{c_1, c_2, \dots, c_q\}$, output a nonnegative integer vector $\tilde{s} \in \mathbb{Z}^q$ such that $\forall \tilde{s}_i \leq c_i$ and $\|\frac{\tilde{s}}{\|\tilde{s}\|_1} - \mathbf{v}\|_1$ is minimized. [37] describes an efficient algorithm to solve **Step 2** in $O(C \times q)$ time, where $C = \sum_{i=1}^q c_i$ and q is the number of nonzero elements in \mathbf{x} . The basic idea is conditioning on the sum of \tilde{s} elements, i.e., $\|\tilde{s}\|_1$. By augmenting with the requirement that $\|\tilde{s}\|_1 = N$, the \tilde{s} minimizing $\|\frac{\tilde{s}}{\|\tilde{s}\|_1} - \mathbf{v}\|_1$ also minimizes $\|\tilde{s} - N\mathbf{v}\|_1$. Then let us set aside the constraints c_i and solve the unconstrained problem. For that, let $U = \sum_{i=1}^q \lceil Nv_i \rceil$ and $L = \sum_{i=1}^q \lfloor Nv_i \rfloor$. If $N \leq L$ (resp. $N \geq U$) then the solution is to set each \tilde{s} value below (resp. above) the corresponding value of Nv_i . If $L < N < U$, then compute $M = N - L$. Let the set \mathcal{L} be the elements of \mathbf{v} having the M largest values of $Nv_i - \lfloor Nv_i \rfloor$. For elements $i \in \mathcal{L}$, set $\tilde{s} = \lceil Nv_i \rceil$. For the other elements, set $\tilde{s} = \lfloor Nv_i \rfloor$. In order to solve the constraint version of the problem, we first fix $\tilde{s} = c_i$ for all entries i such that $c_i < Nv_i$. Then, we solve the unconstrained version of the problem for the remaining elements. For any given N , this yields the optimal \tilde{s} in $O(q)$ time. Since the maximum allowable value of $\|\tilde{s}\|_1$ given the constraints is C , we only need to run this algorithm for each $N \in \{1, \dots, C\}$ to find the optimal value of \tilde{s} .

The running time for **Step 1** is $O(m^3)$ in general. However, NOMP algorithm works column-wise, incremental algorithms exist that lower the complexity of the regression to $O(m^2)$. Combined with the outer loop over m , this brings the overall complexity to $O(m^3)$. In practice, users need a few number of reviews m to be selected for inspecting. Hence, the running time of this step is negligible. As discussed, the running time of **Step 2** is $O(C \times q)$. As $C = |\mathcal{R}_i|$ and

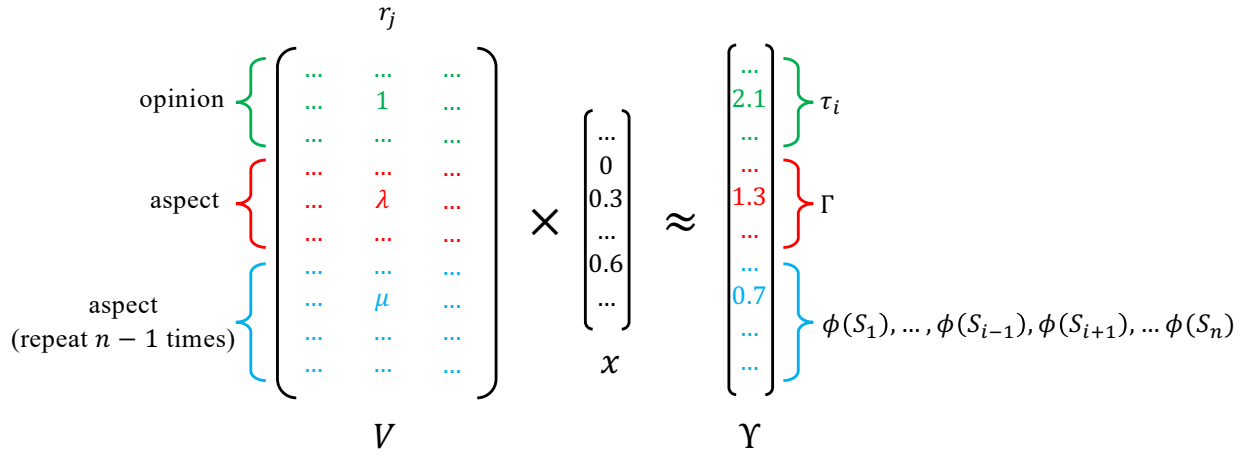


Figure 6.2: A visualization for linear regression of a product p_i when solving COMPARESETS+ $q = O(m)$, the running time of this step is $O(|\mathcal{R}_i| \times m)$. For small value of m , the running time of this step is almost linear to the number of reviews in \mathcal{R}_i .

After we achieve selected sets for each item in the collection of items $\mathcal{P} = \{p_1, p_2, \dots, p_n\}$ by the previously mentioned algorithm, we can incorporate the distance among items for further optimization, that eventually solves the COMPARESETS+ (Equation 6.6) problem simultaneously. In particular, we alternate the following process for each item p_i , for $i = 1$ to n . Let V be a $(m + n \times z) \times |\mathcal{R}_i|$ matrix, in which each entry $V_{ij} = 1$ iff opinion o_i appears in review $r_j \in \mathcal{R}_i$, $V_{(m+i)j} = \lambda$ and $V_{(m+t \times z+i)j} = \mu$ iff aspect $a_{i'}$ appears in review r_j , for $t \in \{1, 2, \dots, n-1\}$. And the target vector for optimization, denoted Υ , is the concatenation of τ_i , Γ , and all other aspect distribution vectors of the selected sets of reviews of other items except item p_i , i.e., $\phi(\mathcal{S}_1), \dots, \phi(\mathcal{S}_{i-1}), \phi(\mathcal{S}_{i+1}), \dots, \phi(\mathcal{S}_n)$. Figure 6.2 visualizes the construction matrix V and target vector Γ . We select a subset reviews for item p_i , finding s by minimizing:

$$\Delta(\Upsilon, Vs) \tag{6.7}$$

We apply same procedure solving Equation 6.4 to solve Equation 6.7. For COMPARESETS+, we perform this algorithm for every item $p_i \in \mathcal{P}$. Thus, the total complexity is $O((m^3 + |\bar{\mathcal{R}}| \times m) \times n)$,

where $|\bar{\mathcal{R}}|$ is the average number of reviews per item.

6.3 Core List of Comparative Items

One insight that we draw is that the objective of COMPARESETS+ effectively captures the pairwise similarities between the comparative items as well as their respective similarity to the target item.

Intuitively, not all comparative items are equally similar. In the event that the initial list of comparative items is long, we may need to narrow it down to a shorter list to make it easier on the end user to read their reviews. Specifically, we are interested in a list of k items that are most similar to each other including the target item.

After solving COMPARESETS+ problem, the distance between item p_i and p_j is:

$$d_{ij} = \Delta(\tau_i, \pi(\mathcal{S}_i)) + \Delta(\tau_j, \pi(\mathcal{S}_j)) + \lambda^2 \Delta(\Gamma, \phi(\mathcal{S}_i)) + \lambda^2 \Delta(\Gamma, \phi(\mathcal{S}_j)) + \mu^2 \Delta(\phi(\mathcal{S}_i), \phi(\mathcal{S}_j)) \quad (6.8)$$

We can construct a complete graph G . Each product is a vertex. Every pair of products p_i and p_j are connected by an edge with a weight $w_{ij} = \max_{p_{i'}, p_{j'} \in \mathcal{P}, i' \neq j'} d_{i'j'} - d_{ij}$ (to turn a notion of distance into similarity). We seek top- k items that are most similar to each other including the target item. This is equivalent to finding the heaviest clique consisting k nodes including p_1 (the target item).

Problem 4. TARGET-ORIENTED HEAVIEST K-SUBGRAPH (TARGETHKS). Given a collection of products $\mathcal{P} = \{p_1, p_2, \dots, p_n\}$, each product p_i is corresponding to a vertex in graph G , the edges are the similarity between any two products (vertices) p_i and p_j is w_{ij} , $i \neq j$. The target product is p_1 . Find a subgraph (subset of products) $\rho \subseteq \mathcal{P}$ such that $|\rho| = k$, $p_1 \in \rho$, and

$$\sum_{p_i, p_j \in \rho, i \neq j} w_{ij} \quad (6.9)$$

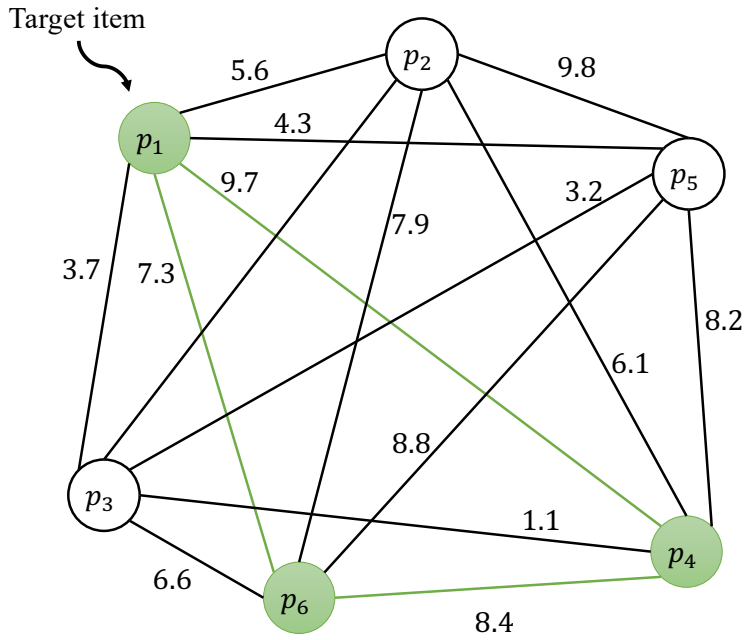


Figure 6.3: An example of target-oriented heaviest 3-subgraph

is maximized.

The TARGETHKS problem is related to finding Heaviest k -Subgraph (HKS) [41]. Our problem is distinct from HKS problem because we seek to find the heaviest k -subgraph that includes the target item. When we solve TARGETHKS with every vertex as the target item, we will eventually find the optimal solution for the HKS problem. Figure 6.3 shows an example of finding target-oriented heaviest 3-subgraph problem. The three products including target product which have the heaviest weight are $\{p_1, p_4, p_6\}$ with a weight of 25.4. Though the set $\{p_2, p_5, p_6\}$ having the heaviest weight of 26.5 is the solution for HKS problem, this solution is excluded from TARGETHKS as it does not include the target item p_1 .

6.3.1 Integer Linear Program

We define the optimal formulation via Integer Linear Programming (ILP). Let γ_i be a binary indicator whether the node p_i is a part of the solution ρ . Objective (6.10a) is the objective to maximize the total weight (similarity) of the selected items. Constraint (6.10b) ensures exactly k items would be selected. Constraint (6.10c) ensures that the target item p_1 is included in the

solution set.

$$\max: \sum_{i=1}^{n-1} \sum_{j=i+1}^n \gamma_i \gamma_j w_{ij} \quad (6.10a)$$

$$\text{s.t: } \sum_{i=1}^n \gamma_i = k \quad (6.10b)$$

$$\gamma_1 = 1 \quad (6.10c)$$

$$\gamma_i \in \{0, 1\}, \forall i \in \{2, 3, \dots, n\} \quad (6.10d)$$

NP-hardness. The TARGETHKS_{ILP} is NP-hard.

Proof. The proof sketch is based on the reduction from vertex cover (known to be NP-hard). Vertex cover finds the minimum set of vertices in a graph, such that all the edges in the graph are covered by at least one of the vertices in this set. We reduce vertex cover to TARGETHKS where $w_{ij} = 1, \forall i \neq j$. Given that the constraints limit γ_i to either 0 or 1, any feasible solution to the TARGETHKS problem is a subset of vertices. If we solve the problem for all $k \in \{1, 2, \dots, n\}$, we arrive at a solution for vertex cover with the minimum set of vertices. \square

6.3.2 Greedy Algorithm

We design a heuristic approach to solve TARGETHKS problem, named TARGETHKS_{Greedy}. Beside being efficient, this algorithm also proves to be effective in practice (see Section 6.4.3). Algorithm 3 iterates $k - 1$ times to select the remaining items excluding the target item. In each iteration, it will loop through the candidate items in which each of them requires to compute the total weight of the current possible solution $\rho \cup \{p_v\}$, which is $O(|\mathcal{P}| \times |\rho|)$. Thus, the total running time of Algorithm 3 is $O((k - 1) \times |\mathcal{P}| \times |\rho|)$.

Algorithm 3 Greedy algorithm: TARGETHKS_{Greedy}

Input: $\mathcal{P}, w_{ij}, k;$

- 1: $\rho = \{p_1\}$
 - 2: **for** $j' = 2 \dots k$ **do**
 - 3: Find an item $p_{i'} \in \mathcal{P}$ that maximizes $\sum_{p_i, p_j \in \rho \cup \{p_{i'}\}, i \neq j} w_{i,j}$
 - 4: $\rho = \rho \cup \{p_{i'}\}$
 - 5: $\mathcal{P} = \mathcal{P} \setminus \{p_{i'}\}$
 - 6: **return** \mathcal{S}_i
-

Dataset	#Product	#Reviewer	#Review	#Target Product	Avg. #Comparison Product	Avg. #Review per Product
Cellphone	10,429	27,879	194,439	9,207	25.57	18.64
Toy	11,924	19,412	167,597	11,004	34.33	14.06
Clothing	23,033	39,387	278,653	21,128	12.03	12.10

Table 6.2: Data statistics

6.4 Experiments

As this is a novel formulation, the experimental objectives are mainly to test the hypothesis that selecting reviews for multiple products jointly result in a better selection for comparative purposes than doing so separately.

6.4.1 Setup

Datasets. We rely on publicly available Amazon Product Review Dataset² [22]. We retrieve comparison products by extracting them from the product metadata in which each product contains a list of “also bought” products for comparison. We conduct experiments with the following categories respectively: Clothing (Clothing), Toys and Games (Toy), Cell Phones and Accessories (Cellphone). Table 6.2 summarizes basic statistics of the datasets. That the average number of comparison items could be as high as 30+ for Toy dataset, motivates why we seek to narrow down to a shorter list.

We acquire sentiment data from [40] in which the authors use a frequency-based approach of [18] to extract aspects from reviews. In particular, using Microsoft Concepts³ as aspects, we

²<http://jmcauley.ucsd.edu/data/amazon/>³<https://concept.research.microsoft.com/>

first retrieve top-2000 most frequently mentioned in reviews, sort them by their correlations with the ratings, and keep only top-500. We may apply other approach to extract aspect sentiment from product reviews. In any case, we consider these as a given.

Baselines. To our best of knowledge, this is the first work on selecting comparative sets of reviews. The closest baseline to the multi-product COMPARESETS and COMPARESETS+ is the single-product CRS[37]⁴. We compare to a heuristic which greedily selects reviews one-by-one such that the selected review minimizes the overall distance cost (i.e., Equation 6.4), named COMPARESETS_{Greedy}. We also compare to Random algorithm, which randomly sample review one-by-one until m reviews have been selected.

Metrics. We measure the alignment of the selected reviews using ROUGE [46], a well-known metric for text matching and text summarization, to assess how well the selected reviews from one item comparing to another item. To cater to words as well as phrases, we report F-score of ROUGE-1 or R-1 (1-gram), ROUGE-2 or R-2 (2-gram), and ROUGE-L or R-L (longest common subsequence). The higher the score, the higher alignment between selected reviews.

Detail Settings. With the availability of data, we investigate COMPARESETS with a setting that the target aspect distribution vector Γ reflects the target item aspect distribution. The maximum number of reviews to be selected $m \in \{3, 5, 10\}$. We tune λ in a candidate set of $\{0.01, 0.1, 1, 10, 100\}$ for COMPARESETS and achieve best performance on ROUGE-L score with $\lambda = 1$. For COMPARESETS+, we set $\lambda = 1$ and tune the coefficient μ in a candidate set of $\{0.01, 0.1, 1, 10, 100\}$ and report the best performance on ROUGE-L score. For TARGETHKS problem, we set $k = m$, the number items k to be selected is the same as the maximum number of reviews m .

⁴We use their best-performing algorithm as baseline. Though that was also based on a form of integer regression, it was significantly different from our method due to the significantly different objectives.

Dataset	Algorithm	$m = 3$					$m = 5$					$m = 10$				
		λ	μ	R-1	R-2	R-L	λ	μ	R-1	R-2	R-L	λ	μ	R-1	R-2	R-L
Cellphone	Random	-	-	15.03	1.12	7.92	-	-	15.02	1.12	7.92	-	-	15.02	1.12	7.92
	CRS	-	-	15.99	1.28	8.44	-	-	15.99	1.28	8.45	-	-	15.92	1.27	8.41
	COMPARESETS _{Greedy}	1	-	15.07	1.12	8.05	1	-	15.08	1.12	8.03	1	-	15.06	1.12	7.98
	COMPARESETS	1	-	16.28	1.36	8.52	1	-	15.99	1.28	8.45	1	-	16.22	1.35	8.47
	COMPARESETS+	1	0.1	16.31 [§]	1.37 [§]	8.72 [§]	1	0.1	16.48 [§]	1.40 [§]	8.74 [§]	1	0.1	16.43 [§]	1.38 [§]	8.67 [§]
Toy	Random	-	-	15.86	1.33	7.98	-	-	15.82	1.32	7.97	-	-	15.84	1.32	7.98
	CRS	-	-	16.26	1.45	8.10	-	-	16.26	1.45	8.12	-	-	16.28	1.45	8.13
	COMPARESETS _{Greedy}	1	-	15.92	1.34	8.13	1	-	15.89	1.33	8.06	1	-	15.87	1.33	8.01
	COMPARESETS	1	-	16.58	1.52	8.28	1	-	16.59	1.52	8.28	1	-	16.57	1.52	8.25
	COMPARESETS+	1	0.1	16.67 [§]	1.54 [§]	8.43 [§]	1	0.1	16.72 [§]	1.55 [§]	8.45 [§]	1	0.1	16.71 [§]	1.55 [§]	8.39 [§]
Clothing	Random	-	-	15.56	1.17	8.46	-	-	15.53	1.17	8.45	-	-	15.54	1.17	8.46
	CRS	-	-	16.37	1.31	8.83	-	-	16.37	1.31	8.83	-	-	16.33	1.32	8.80
	COMPARESETS _{Greedy}	1	-	15.56	1.17	8.52	1	-	15.59	1.17	8.52	1	-	15.57	1.17	8.48
	COMPARESETS	1	-	16.59	1.36 [§]	8.82	1	-	16.58	1.36 [§]	8.82	1	-	16.55	1.36 [§]	8.79
	COMPARESETS+	1	0.1	16.67 [§]	1.36 [§]	8.90 [§]	1	0.1	16.66 [§]	1.36 [§]	8.89 [§]	1	0.1	16.62 [§]	1.36 [§]	8.85 [§]

[§] denotes statistically significant improvements over the second best approach (p -value < 0.05).
Highest values are in **bold**

Table 6.3: Review alignment between target item and comparative items

6.4.2 Comparative Review Sets Selection

Review Alignment Between Target Item and Comparative Items. Here we assess how well the selected sets of reviews of the comparison items align to those of the target item. As reported in Table 6.3, COMPARESETS+ algorithm performs best in ROUGE-L measure across all datasets. COMPARESETS does enhance the alignment of the selected review sets between the target item and comparison items.

Review Alignment Among Comparative Items. Our proposed COMPARESETS+ problem focuses on selecting comparative sets of reviews. Ideally, the selected sets of reviews, when comparing one item to other items, are well-aligned for better comparisons. Table 6.4 show that the proposed COMPARESETS+ consistently outperforms the CRS baselines significantly across all the datasets.

Dataset	Algorithm	$m = 3$			$m = 5$			$m = 10$		
		λ	μ	R-1 R-2 R-L	λ	μ	R-1 R-2 R-L	λ	μ	R-1 R-2 R-L
Cellphone	Random	-	-	14.74 1.05 7.81	-	-	14.73 1.05 7.82	-	-	14.73 1.05 7.81
	CRS	-	-	15.79 1.23 8.40	-	-	15.79 1.23 8.40	-	-	15.69 1.20 8.36
	COMPARESETS _{Greedy}	1	-	15.07 1.12 8.05	1	-	15.08 1.12 8.03	1	-	15.06 1.12 7.98
	COMPARESETS	1	-	15.86 1.23 8.50	1	-	15.86 1.23 8.49	1	-	15.79 1.22 8.44
	COMPARESETS+	1	0.1	15.93[§] 1.24[§] 8.75[§]	1	0.1	16.09[§] 1.28[§] 8.73[§]	1	0.1	16.04[§] 1.26[§] 8.65[§]
Toy	Random	-	-	15.55 1.25 7.90	-	-	15.54 1.25 7.90	-	-	15.54 1.25 7.89
	CRS	-	-	15.97 1.37 8.04	-	-	15.98 1.36 8.07	-	-	15.99 1.37 8.07
	COMPARESETS _{Greedy}	1	-	15.92 1.34 8.13	1	-	15.89 1.33 8.06	1	-	15.87 1.33 8.01
	COMPARESETS	1	-	16.07 1.37 8.25	1	-	16.07 1.37 8.25	1	-	16.06 1.37 8.22
	COMPARESETS+	1	0.1	16.21[§] 1.39[§] 8.40[§]	1	0.1	16.26[§] 1.40[§] 8.41[§]	1	0.1	16.24[§] 1.40[§] 8.35[§]
Clothing	Random	-	-	15.32 1.13 8.37	-	-	15.32 1.13 8.38	-	-	15.32 1.13 8.38
	CRS	-	-	16.18 1.26 8.78	-	-	16.17 1.27 8.77	-	-	16.13 1.27 8.74
	COMPARESETS _{Greedy}	1	-	15.56 1.17 8.52	1	-	15.59 1.17 8.52	1	-	15.57 1.17 8.48
	COMPARESETS	1	-	16.12 1.25 8.74	1	-	16.12 1.26 8.74	1	-	16.10 1.25 8.71
	COMPARESETS+	1	0.1	16.20[§] 1.25 8.82[§]	1	0.1	16.20[§] 1.25 8.81[§]	1	0.1	16.17[§] 1.26 8.77[§]

[§] denotes statistically significant improvements over the second best approach (p -value < 0.05).
Highest values are in **bold**

Table 6.4: Comparison to baselines in review alignment among comparative items

6.4.3 Core List of Comparative Items

Optimal vs Approximation. For the optimal $\text{TARGETHKS}_{\text{ILP}}$, when limited to 60 seconds, the Gurobi⁵ solver could still solve optimally for virtually all problem instances for smaller values of $k \in \{3, 5\}$. For the larger $k = 10$, the percentages vary from two-thirds to close to full across datasets.

We are then interested in seeing how well the approximation could approach the optimal. Table 6.5 reports the ratios of the difference between the objective values achieved by approximation algorithms and the optimal for $\text{TARGETHKS}_{\text{ILP}}$,

$$\text{Objective Value Ratio} = \frac{\text{Objective}_{\text{approximation}} - \text{Objective}_{\text{TargetHkS}_{\text{ILP}}}}{\text{Objective}_{\text{TargetHkS}_{\text{ILP}}}} \quad (6.11)$$

The heuristic approach $\text{TARGETHKS}_{\text{Greedy}}$ achieves quite good objective values that almost similar to those of $\text{TARGETHKS}_{\text{ILP}}$. Random approach performs poorly with a big gap, with objective value reduction ratio $> 20\%$ overall.

⁵<https://www.gurobi.com/>

Dataset	k	# Optimal Solution	Objective Value Ratio	
			TARGETHKS _{Greedy}	Random
Cellphone	3	100.00	-0.00005	-21.97
	5	99.59	-0.00008	-22.14
	10	80.85	-0.00002	-18.96
Toy	3	100.00	-0.00009	-19.39
	5	99.00	-0.00015	-21.04
	10	66.78	0.00147	-20.14
Clothing	3	100.00	-0.00013	-24.59
	5	99.98	-0.00013	-23.28
	10	98.45	-0.00004	-18.00

Table 6.5: Performance ratios over TARGETHKS_{ILP} (%)

Dataset	Algorithm	$k = m = 3$			$k = m = 5$			$k = m = 10$		
		R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
Cellphone	Random	16.34	1.39	8.74	16.49	1.39	8.74	16.42	1.39	8.67
	TARGETHKS _{Greedy}	16.91	1.54	8.91	16.88	1.50	8.93	16.65	1.44	8.79
	TARGETHKS _{ILP}	16.93	1.54	8.93	16.89	1.50	8.93	16.64	1.44	8.78
Toy	Random	16.72	1.56	8.46	16.72	1.56	8.46	16.70	1.54	8.38
	TARGETHKS _{Greedy}	17.11	1.67	8.67	16.96	1.62	8.61	16.84	1.57	8.48
	TARGETHKS _{ILP}	17.14	1.71	8.70	17.00	1.64	8.63	16.85	1.58	8.48
Clothing	Random	16.67	1.37	8.90	16.69	1.37	8.90	16.62	1.36	8.84
	TARGETHKS _{Greedy}	16.91	1.41	9.02	16.78	1.38	8.95	16.66	1.37	8.87
	TARGETHKS _{ILP}	16.94	1.42	9.03	16.80	1.39	8.96	16.66	1.37	8.87

Highest values are in **bold**

Table 6.6: Review alignment between target item and comparison items for core list of comparative items

Review Alignment. We assess the similarity between target item with comparison items by measuring ROUGE score between the reviews selected by the items in the top- k similar items selected by the approximation algorithm against those of TARGETHKS_{ILP} (see Table 6.6). We also assess the similarity among all items (see Table 6.7). The TARGETHKS_{Greedy}'s performance approaches those of TARGETHKS_{ILP}.

6.4.4 Case Study

For illustration, Figure 6.4 shows a case study of Car Charger products. These three products are the top-3 most similar items selected from the list of total 9 also bought products by

Dataset	Algorithm	$k = m = 3$			$k = m = 5$			$k = m = 10$		
		R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
Cellphone	Random	16.20	1.33	8.75	16.20	1.30	8.72	16.06	1.27	8.65
	TARGETHKS _{Greedy}	16.95	1.53	9.02	16.77	1.45	9.03	16.38	1.35	8.84
	TARGETHKS _{ILP}	16.96	1.53	9.04	16.78	1.45	9.03	16.38	1.35	8.84
Toy	Random	16.53	1.49	8.44	16.41	1.45	8.44	16.28	1.41	8.35
	TARGETHKS _{Greedy}	17.12	1.67	8.78	16.91	1.60	8.76	16.61	1.48	8.59
	TARGETHKS _{ILP}	17.16	1.70	8.80	16.95	1.61	8.79	16.62	1.49	8.60
Clothing	Random	16.46	1.31	8.87	16.32	1.28	8.84	16.19	1.26	8.77
	TARGETHKS _{Greedy}	16.79	1.38	9.04	16.53	1.32	8.98	16.27	1.27	8.85
	TARGETHKS _{ILP}	16.82	1.38	9.06	16.55	1.32	8.99	16.27	1.27	8.85

Highest values are in **bold**

Table 6.7: Review alignment among items for core list of comparative items

TARGETHKS_{ILP}. Every product has 3 reviews. All of them discuss about a common aspect *charger*. The selected set of reviews for each product cover diverse sentiments on various aspects that are related. One aspect they touch on is the use of the charger for iPhone (mentioned in the selected reviews for all products). Another aspect is its in-car use (also universally mentioned). The second review of the first product and the third review of the third product discuss durability. The third review of the first product and the first review of the third product discuss how quickly it charges. This shows how synchronizing the review selection across comparative products help to feature reviews that allow better comparisons.

6.4.5 User Study

To conduct a qualitative study on the efficacy of the sets of reviews from human perspective, we select 3 examples from each category to get 9 examples in total and design 3 independent surveys, each containing 9 examples of different review selection algorithms (presented blindly), involving 15 participants who are not the authors (each example is assessed by 5 participants). Each example contains a target product and 2 other products which are most relevant selected by TARGETHKS_{ILP}. For parity, we only select examples which have 3 selected reviews from COMPARESETS+, CRS, and Random algorithms. There are three questions:

Q1: How similar are the reviews among products (i.e., discussing same aspects)?

Compare to similar items

This item: Skiva PowerFlow 2.1Amp / 10Watt (Fast) Car Charger (Now with Improved Cable) for new iPad, iPhone 4S 4 3GS, iPad 2, iPad 3, iPhone, iPad, & iPod



★★★★★

A nice 2.1 Amp charger for the car that doesn't cost \$30. Got this as a Christmas present for my dad to use with his iPhone, and no issues thus far.

★★★★☆

This wasn't the fastest charger but definitely worked for about a month. The cord must be cheaply made however as it stopped working after a month. I kept the car charge plug in piece but haven't had the chance to test it with a new cord.

★★★★★

This is the best charger I have ever had. It charges quickly and faster than my Kingston rapid charger. Really happy with this and purchased an additional one for my wife.

Belkin Car Charger with Lightning Cable Connector to USB Cable for iPhone 5 / 5S / 5c, iPad (4th Gen), iPad mini, iPod touch (5th Gen), and iPod nano (7th Gen) (2.1 AMP / 10 Watt)



★★★★★

This is exactly what I expected! I needed a charger for my iPhone and this is the one apple recommended. Works great

★★★★★

I needed a car charger and this one works well. I keep it in the car in case my phone needs charging.

★★★★★

Seems like the original product, not a copy. Bought from Amazon. Arrived quickly. Just as described. I'm very satisfied with the cable and the USB charger. Thanks.

Cbus Wireless Vehicle Car Charger for Apple iPad / iPad 2 / iPad 3 / iPhone 4S / iPhone 4 / iPhone 3G / iPhone 3Gs / iPod Touch 4 / 4G / 4th / 3rd / 2nd Gen.



★★★★★

I love this charger. It works awesome for my iPhone 4s and it charges the phone pretty quickly. Great product for the price.

★★★★★

I needed this charger for my car, it works well for me. It charges my phone quickly. I recommend this

★★★★☆

I used this charger for a while and then it stopped working, I had to be moving it around for it to work. #THESTRUGGLE. not worth it :(

Figure 6.4: Example selected sets of reviews of a Cellphone instance

Algorithm	Q1	Q2	Q3
Random	3.47	3.78	3.38
CRS	3.69	4.07	3.64
COMPARESETS+	3.73	4.18	3.71

Highest values are in **bold**

Table 6.8: Result analysis of user study

Q2: Do reviews help you know more about the recommended products?

Q3: Do reviews help you in comparison among products?

The first looks into similar aspects among reviews selected from different algorithms. The second looks into the appropriateness of the selected reviews with the product. The third looks into the comparative information given by the selected sets of reviews for comparing the products. Each participant chooses from five-point Likert scale, from 1 (strongly disagree/strongly dissimilar) to 5 (strongly agree/strongly similar).

The overall result is reported in Table 6.8. Given the abundant of reviews for each product, the average scores are > 3 , which indicate the reviews are provide useful information for the appropriate products. The evaluation scores of COMPARESETS+ are consistently the highest among comparison algorithms. The narrowing down list of products produced by TARGETHKS_{ILP} are quite similar. This explains the small gap of Q1 scores between CRS and COMPARESETS+. The higher scores on Q2 may indicate that most of the reviews from all algorithms are informative to provide user with additional information to know more about the recommendation product. For Q3, there is an improvement in comparative information in reviews selected by COMPARESETS+ among products.

6.5 Summary

In this chapter, we address a novel problem of selecting comparative sets of reviews for a set of comparable items, which include one target item and other comparison items. The review selection process can be performed individually (COMPARESETS) or synchronously (COMPARESETS+). After the review selection process, we narrow down the list of comparable items to top- k most similar items including the target item. The experiment results validate the efficacies of our proposed heuristic algorithms.

Chapter 7

Conclusion

7.1 Summary

In this dissertation, we present our research on mining product textual data for recommendation explanations. We formulate and propose effective solutions covering various forms of textual recommendation explanation, where they are produced jointly or separately.

We propose an innovative post hoc strategy for providing natural language explanations for personalized recommendations in Chapter 3. Our approach synthesizes an explanation by selecting representative sentences from a product’s reviews, contextualizing the opinions based on aspect-level sentiments from a class of compatible explainable recommendation models. Although relying on inputs from another model is a main limitation of SEER, this allows SEER to be flexible to adapt with different models. Experiments on five different product categories showcase the efficacies of our method as compared to baselines based on templates, review summarization, selection, and text generation.

In Chapter 4, we propose an attention neural network model named QUESTER that employs QA in an attention mechanism that aligns reviews to various QAs of an item and assesses their contribution jointly to the recommendation objective. The proposed model improves *review-level explanation* as the QA aids in selecting more useful reviews. The accompanied QA with the

well-aligned review also play as an expanded form of explanation. Experiments on ten different product categories showcase the efficacies of QUESTER to comparable baselines in identifying useful reviews and QAs, while maintaining parity in recommendation performance.

In Chapter 5, we develop an explainable recommendation model that produces comparative explanations jointly with the recommended results. Not only does the model aim at providing comparative explanation from the given item with respect to another reference item, but we also formulate comparative constraints involving aspect-level comparisons between the target item and reference item. Experiments on four public datasets of several product categories showcase the efficacies of our methodology as compared to baselines at attaining better recommendation accuracies and intuitive explanations.

In Chapter 6, we tackle selecting comparative sets of reviews, which is also a novel form of recommendation explanation that cater information from multiple reviews of comparable products for ease of comparison when a user viewing a product in comparison to other similar products. We formulate the objective function that allow us to select sets of reviews from multiple products simultaneously. After review selection process, we further reduce the number of products to top- k most similar products by formulating the set of comparable products as a complete graph and find a k -subgraph (k products) that maximizes the similarity among products, this subgraph always include the target product. We proposed efficient approximation algorithms to solve the proposed objective. Experiments on various product categories validate efficacies of our method.

7.2 Future Research

This dissertation broadly covers various form of recommendation explanations. We identify and propose a few potential research directions as our future work.

7.2.1 Improving Aspect-Level Sentiment Sentences within Explanation

Chapter 3 discussed our proposed framework SEER, a post-hoc method for synthesizing explanation for recommended item from another compatible explainable recommendation model. In the objective, we address the representativeness of the selected sentence to other sentences in the candidates from item reviews and the coherence by constructed from a cost function of the aspect-level preference of the target user and the reviewers of reviews. The tradeoff between the coherence and representativeness has not been investigate. Furthermore, the opinion substitution model decouples the sentiment from the review selection process, making the two-step framework. We would like to extend the objective of SEER to incorporate the sentiment of each sentence.

Sentiment. The aspect sentiment of the selected sentence should reflect the inferred aspect sentiment of the target user preference. Based on the aspect sentiment scores, we can measure the distance between the sentiment of the target user

$$\text{s_cost}(\tau) = \sum_{s \in \mathcal{S}_j} \rho_s \cdot \gamma_s \quad (7.1)$$

Overall Cost. The overall cost is thus:

$$\text{cost}(\tau) = \text{c_cost}(\tau) + \alpha \cdot \text{r_cost}(\tau) + \mu \cdot \text{s_cost}(\tau) \quad (7.2)$$

We introduce α and μ to control the contribution of the representativeness and sentimental cost.

$$\min: \sum_{t_{i'j} \in \mathcal{T}_j} \zeta_{i'} \cdot \theta_{i'} + \alpha \sum_{s \in \mathcal{S}_j} \sum_{s' \in \mathcal{S}_j} \Gamma_{ss'} \cdot \delta_{ss'} + \mu \sum_{s \in \mathcal{S}_j} \rho_s \cdot \gamma_s \quad (7.3a)$$

$$\text{s.t.} \sum_{s \in \mathcal{S}_j} \Gamma_{ss'} = 1, \forall s' \in \mathcal{S}_j \quad (7.3b)$$

$$\Gamma_{ss'} \leq \gamma_s, \forall s, s' \in \mathcal{S}_j \quad (7.3c)$$

$$\gamma_s \cdot \sigma_{si'} \leq \zeta_{i'}, \forall t_{i'j} \in \mathcal{T}_j, s \in \mathcal{S}_j \quad (7.3d)$$

$$\Gamma_{ss'} \leq \sum_{a_k \in \mathcal{A}} \pi_{sk} \cdot \pi_{s'k}, \forall s, s' \in \mathcal{S}_j \quad (7.3e)$$

$$\sum_{s \in \mathcal{S}_j} \gamma_s \cdot \pi_{sk} = \mathcal{D}_k, \forall a_k \in \mathcal{A} \quad (7.3f)$$

$$\zeta_{i'}, \gamma_s, \Gamma_{ss'} \in \{0, 1\}, \forall t_{i'j} \in \mathcal{T}_j; \forall s, s' \in \mathcal{S}_j \quad (7.3g)$$

α is the trace-off factor controlling the contribution of representative cost and μ is the trace-off factor controlling the contribution of sentimental cost.

7.2.2 Modeling Subjective and Objective Aspect-Level Quality Jointly

In Chapter 5, we validated the efficacies of our proposed model COMPAREER on both subjective and objective modes of aspect-level quality. However, in practice, both subjective and objective modes may appear simultaneously, which motivates us in modeling these two modes jointly. Although this is a promising direction, one challenge is that the data for such scenario is not yet available. We can also explore comparisons to other definitions of reference items such as substitutes, incorporating the comparative constraints into other recommendation models/objectives, testing multi-way comparisons to multiple products simultaneously, etc.

Bibliography

- [1] Banerjee, S., and Lavie, A. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, 65–72. 4.2.1
- [2] Barrios, F.; López, F.; Argerich, L.; and Wachenchauser, R. 2016. Variations of the similarity function of textrank for automated summarization. *CoRR* abs/1602.03606. 1.2, 2.1, 2.3.4, 3.5.3
- [3] Bauman, K.; Liu, B.; and Tuzhilin, A. 2017. Aspect based recommendations: Recommending items with the most valuable aspects based on user reviews. In *KDD, KDD’17*, 717–725. ACM. 2.1, 3.4
- [4] Borzsony, S.; Kossmann, D.; and Stocker, K. 2001. The skyline operator. In *ICDE*, 421–430. IEEE. 5
- [5] Chen, L., and Wang, F. 2017. Explaining recommendations based on feature sentiments in product reviews. In *IUI, IUI’17*, 17–28. ACM. 1.3, 2.4
- [6] Chen, X.; Qin, Z.; Zhang, Y.; and Xu, T. 2016. Learning to rank features for recommendation over multiple categories. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR’16*, 305–314. New York, NY, USA: Association for Computing Machinery. 2.1, 3.4
- [7] Chen, Q.; Zhu, X.; Ling, Z.-H.; Wei, S.; Jiang, H.; and Inkpen, D. 2017. Enhanced LSTM for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1657–1668. Vancouver, Canada: Association for Computational Linguistics. 3.5.1
- [8] Chen, C.; Zhang, M.; Liu, Y.; and Ma, S. 2018. Neural attentional rating regression with review-level explanations. In *WWW, WWW’18*, 1583–1592. 1.2, 2.3.3, 4, 4.1, 4.1, 4.2, 4.2
- [9] Chen, X.; Chen, H.; Xu, H.; Zhang, Y.; Cao, Y.; Qin, Z.; and Zha, H. 2019a. Personalized fashion recommendation with visual explanations based on multimodal attention network: Towards visually explainable recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR’19*, 765–774. New York, NY, USA: Association for Computing Machinery. (document), 2.1, 2.2.2, 2.5
- [10] Chen, Z.; Wang, X.; Xie, X.; Wu, T.; Bu, G.; Wang, Y.; and Chen, E. 2019b. Co-attentive multi-task learning for explainable recommendation. In *Proceedings of the Twenty-Eighth*

International Joint Conference on Artificial Intelligence, IJCAI-19, 2137–2143. International Joint Conferences on Artificial Intelligence Organization. 2.3.5

- [11] Chen, L.; Guan, Z.; Xu, Q.; Zhang, Q.; Sun, H.; Lu, G.; and Cai, D. 2020. Question-driven purchasing propensity analysis for recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 35–42. 4
- [12] Cong, D.; Zhao, Y.; Qin, B.; Han, Y.; Zhang, M.; Liu, A.; and Chen, N. 2019. Hierarchical attention based neural network for explainable recommendation. In *Proceedings of the 2019 on International Conference on Multimedia Retrieval, ICMR '19*, 373–381. New York, NY, USA: Association for Computing Machinery. (document), 2.2.2, 2.6, 2.3.3
- [13] Cornuéjols, G.; Nemhauser, G. L.; and Wolsey, L. A. 1983. The uncapacitated facility location problem. Technical report, Carnegie-mellon univ pittsburgh pa management sciences research group. 3.2.2
- [14] Diao, Q.; Qiu, M.; Wu, C.-Y.; Smola, A. J.; Jiang, J.; and Wang, C. 2014. Jointly modeling aspects, ratings and sentiments for movie recommendation (jmars). In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14*, 193–202. New York, NY, USA: ACM. 4
- [15] Diaz, G. O., and Ng, V. 2018. Modeling and prediction of online product review helpfulness: a survey. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 698–708. 2.3.3
- [16] Dong, L.; Huang, S.; Wei, F.; Lapata, M.; Zhou, M.; and Xu, K. 2017. Learning to generate product reviews from attributes. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, volume 1, 623–632. 1.2, 2.1, 2.3.5, 3.5.3
- [17] Fan, M.; Feng, C.; Guo, L.; Sun, M.; and Li, P. 2019. Product-aware helpfulness prediction of online reviews. In *The World Wide Web Conference, WWW '19*, 2715–2721. New York, NY, USA: Association for Computing Machinery. 2.3.3
- [18] Gao, J.; Wang, X.; Wang, Y.; and Xie, X. 2019. Explainable recommendation through attentive multi-view learning. In *AAAI*, 3622–3629. AAAI Press. 2.1, 2.3.2, 3.4, 5.4.1, 6.4.1
- [19] Ghose, A., and Ipeirotis, P. G. 2007. Designing novel review ranking systems: predicting the usefulness and impact of reviews. In *Proceedings of the ninth international conference on Electronic commerce*, 303–310. 2.3.3
- [20] Golab, L.; Korn, F.; Li, F.; Saha, B.; and Srivastava, D. 2015. Size-constrained weighted set cover. In *2015 IEEE 31st International Conference on Data Engineering*, 879–890. IEEE. 3.2.3
- [21] Hada, D. V.; M., V.; and Shevade, S. K. 2021. Rexplug: Explainable recommendation using plug-and-play language model. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '21*, 81–91. 2.3.5
- [22] He, R., and McAuley, J. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *WWW, WWW'16*, 507–517. 4.2, 5.1, 6.4.1
- [23] He, X.; Chen, T.; Kan, M.-Y.; and Chen, X. 2015. Trirank: Review-aware explainable

- recommendation by modeling aspects. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM'15, 1661–1670. New York, NY, USA: Association for Computing Machinery. 2.1, 3.4
- [24] Herlocker, J. L.; Konstan, J. A.; Terveen, L. G.; and Riedl, J. T. 2004. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.* 22(1):5–53. (document), 1.1, 1.2, 2.3, 2.3.1, 4
- [25] Hochbaum, D. S. 1982. Heuristics for the fixed cost median problem. *Mathematical programming* 22(1):148–162. 3.2.3
- [26] Hu, M., and Liu, B. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 168–177. 2.3.4, 5.2, 5.4.1
- [27] Huang, C.; Jiang, W.; Wu, J.; and Wang, G. 2020. Personalized review recommendation based on users' aspect sentiment. *ACM Trans. Internet Technol.* 20(4). 2.3.3
- [28] Jang, M.; Park, J.-w.; and Hwang, S.-w. 2012. Predictive mining of comparable entities from the web. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*. 2.4
- [29] Kim, S.-M.; Pantel, P.; Chklovski, T.; and Pennacchiotti, M. 2006. Automatically assessing review helpfulness. In *Proceedings of the 2006 Conference on empirical methods in natural language processing*, 423–430. 2.3.3
- [30] Kingma, D. P., and Ba, J. 2015. Adam: A method for stochastic optimization. In *ICLR (Poster)*. 4.2
- [31] Kolda, T. G., and Bader, B. W. 2009. Tensor decompositions and applications. *SIAM review* 51(3):455–500. 3.4, 5.3.1
- [32] Komwad, N.; Tiwari, P.; Praveen, B.; and Chowdary, C. R. 2022. A survey on review summarization and sentiment classification. *Knowledge and Information Systems* 1–39. 2.3.4
- [33] Koren, Y.; Bell, R.; and Volinsky, C. 2009. Matrix factorization techniques for recommender systems. *Computer* (8):30–37. 1.1, 4
- [34] Lan, W., and Xu, W. 2018. Neural network models for paraphrase identification, semantic textual similarity, natural language inference, and question answering. In *Proceedings of COLING 2018*, 3890–3902. 3.2.1
- [35] Landis, J. R., and Koch, G. G. 1977. The measurement of observer agreement for categorical data. *Biometrics* 33(1):159–174. 4.2.6
- [36] Lappas, T., and Gunopulos, D. 2010. Efficient confident search in large review corpora. In *Joint European conference on machine learning and knowledge discovery in databases*, 195–210. Springer. 2.3.3
- [37] Lappas, T.; Crovella, M.; and Terzi, E. 2012. Selecting a characteristic set of reviews. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 832–840. ACM. 2.3.3, 3.5.3, 6, 6.1, 6.1, 6.2.2, 6.4.1
- [38] Lappas, T.; Valkanas, G.; and Gunopulos, D. 2012. Efficient and domain-invariant competitor mining. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge*

discovery and data mining, 408–416. 2.4

- [39] Le, T.-H., and Lauw, H. W. 2020. Synthesizing aspect-driven recommendation explanations from reviews. In Bessiere, C., ed., *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, 2427–2434. International Joint Conferences on Artificial Intelligence Organization. Main track. (document), 1.4, 2.1
- [40] Le, T.-H., and Lauw, H. W. 2021. Explainable recommendation with comparative constraints on product aspects. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining, WSDM '21*, 967–975. New York, NY, USA: Association for Computing Machinery. (document), 1.3, 1.4, 2.3.2, 2.4, 6.4.1
- [41] Letsios, M.; Balalau, O. D.; Danisch, M.; Orsini, E.; and Sozio, M. 2016. Finding heaviest k-subgraphs and events in social media. In *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, 113–120. IEEE. 6.3
- [42] Li, S.; Zha, Z.-J.; Ming, Z.; Wang, M.; Chua, T.-S.; Guo, J.; and Xu, W. 2011. Product comparison using comparative relations. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'11*, 1151–1152. New York, NY, USA: Association for Computing Machinery. 2.4
- [43] Li, P.; Wang, Z.; Ren, Z.; Bing, L.; and Lam, W. 2017. Neural rating regression with abstractive tips generation for recommendation. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'17*, 345–354. New York, NY, USA: Association for Computing Machinery. 1.2, 2.3.4, 2.3.5
- [44] Li, L.; Zhang, Y.; and Chen, L. 2020. Generate neural template explanations for recommendation. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management, CIKM '20*, 755–764. 2.3.5
- [45] Li, L.; Zhang, Y.; and Chen, L. 2021. Personalized transformer for explainable recommendation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 4947–4957. 2.3.5
- [46] Lin, C.-Y., and Hovy, E. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *NAACL*. 3.5, 4.2.1, 6.4.1
- [47] Lipton, Z. C. 2018. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue* 16(3):31–57. 2.1
- [48] Liu, J.; Cao, Y.; Lin, C.-Y.; Huang, Y.; and Zhou, M. 2007. Low-quality product review detection in opinion summarization. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, 334–342. 2.3.3
- [49] Liu, Y.; Huang, X.; An, A.; and Yu, X. 2008. Modeling and predicting the helpfulness of online reviews. In *2008 Eighth IEEE international conference on data mining*, 443–452. IEEE. 2.3.3
- [50] Liu, J.; Xiong, L.; Pei, J.; Luo, J.; Zhang, H.; and Zhang, S. 2019. Skyrec: Finding pareto optimal groups. In *CIKM, CIKM'19*, 2913–2916. ACM. 5

- [51] Liu, H.; Wang, Y.; Peng, Q.; Wu, F.; Gan, L.; Pan, L.; and Jiao, P. 2020. Hybrid neural recommendation with joint deep representation learning of ratings and reviews. *Neurocomputing* 374:77–85. 2.3.3, 4, 4.1, 4.1, 4.1, 4.2, 4.2
- [52] Lu, Y.; Tsaparas, P.; Ntoulas, A.; and Polanyi, L. 2010. Exploiting social context for review quality prediction. In *Proceedings of the 19th international conference on World wide web*, 691–700. 2.3.3
- [53] Lu, Y.; Dong, R.; and Smyth, B. 2018. Why i like it: Multi-task learning for recommendation and explanation. In *RecSys, RecSys’18*, 4–12. ACM. 2.3.5
- [54] Martin, L., and Pu, P. 2014. Prediction of helpful reviews using emotions extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28. 2.3.3
- [55] McAuley, J., and Leskovec, J. 2013. Hidden factors and hidden topics: Understanding rating dimensions with review text. In *Proceedings of the 7th ACM Conference on Recommender Systems, RecSys’13*, 165–172. New York, NY, USA: Association for Computing Machinery. (document), 2.1, 2.3.3, 4.2
- [56] McAuley, J.; Targett, C.; Shi, Q.; and van den Hengel, A. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR ’15*, 43–52. New York, NY, USA: ACM. 3.5
- [57] McAuley, J.; Pandey, R.; and Leskovec, J. 2015. Inferring networks of substitutable and complementary products. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD’15*, 785–794. New York, NY, USA: Association for Computing Machinery. 2.4
- [58] Melamud, O.; Goldberger, J.; and Dagan, I. 2016. context2vec: Learning generic context embedding with bidirectional lstm. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, 51–61. 3.3, 3.5.2
- [59] Meng, X., and Wang, H. 2009. Mining user reviews: from specification to summarization. In *Proceedings of the ACL-IJCNLP 2009 conference short papers*, 177–180. 2.3.4
- [60] Nguyen, T.-S.; Lauw, H. W.; and Tsaparas, P. 2015. Review synthesis for micro-review summarization. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, WSDM ’15*, 169–178. New York, NY, USA: ACM. 2.1
- [61] Ni, J., and McAuley, J. 2018. Personalized review generation by expanding phrases and attending on aspect-aware representations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, 706–711. 2.3.5, 3.5.3
- [62] Ni, J.; Li, J.; and McAuley, J. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 188–197. ACL. 2.3.5, 3.5.3
- [63] Nie, Y., and Bansal, M. 2017. Shortcut-stacked sentence encoders for multi-domain inference. In *Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for*

- NLP*, 41–45. Copenhagen, Denmark: Association for Computational Linguistics. 3.5.1
- [64] Peake, G., and Wang, J. 2018. Explanation mining: Post hoc interpretability of latent factor models for recommendation systems. In *KDD, KDD'18*, 2060–2069. ACM. (document), 2.1
- [65] Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543. 4.2
- [66] Pu, P.; Chen, L.; and Hu, R. 2011. A user-centric evaluation framework for recommender systems. In *Proceedings of the fifth ACM conference on Recommender systems*, 157–164. 4
- [67] Ren, Z.; Liang, S.; Li, P.; Wang, S.; and de Rijke, M. 2017. Social collaborative viewpoint regression with explainable recommendations. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, WSDM'17*, 485–494. New York, NY, USA: Association for Computing Machinery. (document), 2.1, 4
- [68] Rendle, S.; Freudenthaler, C.; Gantner, Z.; and Schmidt-Thieme, L. 2009. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI'09*, 452–461. Arlington, Virginia, USA: AUAI Press. 5.3.1
- [69] Sachdeva, N., and McAuley, J. 2020. *How Useful Are Reviews for Recommendation? A Critical Review and Potential Improvements*. Association for Computing Machinery. 1845–1848. 4.2.4
- [70] Saumya, S.; Singh, J. P.; and Dwivedi, Y. K. 2020. Predicting the helpfulness score of online reviews using convolutional neural network. *Soft Computing* 24(15):10989–11005. 2.3.3
- [71] Shimada, K.; Tadano, R.; and Endo, T. 2011. Multi-aspects review summarization with objective information. *Procedia-Social and Behavioral Sciences* 27:140–149. 2.3.4
- [72] Tai, C.-Y.; Huang, L.-Y.; Huang, C.-K.; and Ku, L.-W. 2021. User-centric path reasoning towards explainable recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '21*, 879–889. New York, NY, USA: Association for Computing Machinery. 2.1
- [73] Tan, Y.; Zhang, M.; Liu, Y.; and Ma, S. 2016. Rating-boosted latent topics: Understanding users and items with ratings and reviews. In *IJCAI*, volume 16, 2640–2646. 2.1, 4, 4.1
- [74] Tang, J.; Gao, H.; Hu, X.; and Liu, H. 2013. Context-aware review helpfulness rating prediction. In *Proceedings of the 7th ACM Conference on Recommender Systems, RecSys '13*, 1–8. 2.3.3
- [75] Tao, Y.; Jia, Y.; Wang, N.; and Wang, H. 2019. The fact: Taming latent factor models for explainability with factorization trees. In *SIGIR, SIGIR'19*, 295–304. ACM. 2.1, 2.3.2
- [76] Tay, Y.; Luu, A. T.; and Hui, S. C. 2018. Multi-pointer co-attention networks for recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '18*, 2309–2318. 4
- [77] Tintarev, N., and Masthoff, J. 2015. *Explaining Recommendations: Design and Evaluation*. Boston, MA: Springer US. 353–382. 1.2, 2.3.1

- [78] Tkachenko, M., and Lauw, H. W. 2014. Generative modeling of entity comparisons in text. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM'14*, 859–868. Association for Computing Machinery. 2.4
- [79] Truong, Q.-T., and Lauw, H. W. 2017. Visual sentiment analysis for review images with item-oriented and user-oriented cnn. In *Proceedings of the 25th ACM International Conference on Multimedia, MM '17*, 1274–1282. New York, NY, USA: Association for Computing Machinery. (document), 2.1, 2.2.3, 2.7
- [80] Truong, Q.-T., and Lauw, H. 2019a. Multimodal review generation for recommender systems. In *WWW, WWW'19*, 1864–1874. ACM. (document), 2.8, 2.2.4, 2.3.5
- [81] Truong, Q.-T., and Lauw, H. W. 2019b. Vistanet: Visual aspect attention network for multimodal sentiment analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 305–312. 2.2.4
- [82] Tsaparas, P.; Ntoulas, A.; and Terzi, E. 2011. Selecting a comprehensive set of reviews. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '11*, 168–176. New York, NY, USA: ACM. 2.3.3, 3.5.3
- [83] Tsur, O., and Rappoport, A. 2009. Revrank: A fully unsupervised algorithm for selecting the most helpful book reviews. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 3, 154–161. 2.3.3
- [84] Wang, C., and Blei, D. M. 2011. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '11*, 448–456. New York, NY, USA: ACM. 2.1
- [85] Wang, N.; Wang, H.; Jia, Y.; and Yin, Y. 2018a. Explainable recommendation via multi-task learning in opinionated text data. In *SIGIR, SIGIR'18*, 165–174. ACM. 1.2, 1.4, 2.1, 2.3.2, 3, 3.4, 3.5.3, 3.5.4, 5, 5, 5.3.1, 5.3.1, 5.4.1, 5.5.2
- [86] Wang, X.; Chen, Y.; Yang, J.; Wu, L.; Wu, Z.; and Xie, X. 2018b. A reinforcement learning framework for explainable recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*, 587–596. IEEE. 2.3.4
- [87] Wang, X.; Wang, D.; Xu, C.; He, X.; Cao, Y.; and Chua, T.-S. 2019. Explainable reasoning over knowledge graphs for recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 5329–5336. 2.1
- [88] Wu, Y., and Ester, M. 2015. Flame: A probabilistic model combining aspect based opinion mining and collaborative filtering. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, WSDM'15*, 199–208. New York, NY, USA: Association for Computing Machinery. (document), 1.2, 2.1, 2.2.2, 2.4, 3.4
- [89] Xu, N.; Liu, H.; Chen, J.; He, J.; and Du, X. 2014. *Selecting a Representative Set of Diverse Quality Reviews Automatically*. 488–496. 2.3.3
- [90] Yang, Y.; Tang, J.; Keomany, J.; Zhao, Y.; Li, J.; Ding, Y.; Li, T.; and Wang, L. 2012. Mining competitive relationships by learning across heterogeneous networks. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, 1432–1441. 2.4

- [91] Yu, Q., and Lam, W. 2018. Review-aware answer prediction for product-related questions incorporating aspects. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, WSDM '18, 691–699. 4
- [92] Yu, W.; Zhang, R.; He, X.; and Sha, C. 2013. Selecting a diversified set of reviews. In *Asia-Pacific Web Conference*, 721–733. Springer. 2.3.3
- [93] Zhan, J.; Loh, H. T.; and Liu, Y. 2009. Gather customer concerns from online product reviews—a text summarization approach. *Expert Systems with Applications* 36(2):2107–2115. 2.3.4
- [94] Zhang, Y., and Chen, X. 2020. Explainable recommendation: A survey and new perspectives. *Foundations and Trends in Information Retrieval* 14(1):1–101. 4
- [95] Zhang, Z., and Varadarajan, B. 2006. Utility scoring of product reviews. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, 51–57. 2.3.3
- [96] Zhang, Y.; Lai, G.; Zhang, M.; Zhang, Y.; Liu, Y.; and Ma, S. 2014. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'14, 83–92. New York, NY, USA: Association for Computing Machinery. 1.1, 1.2, 1.4, 2.1, 2.3.2, 3, 3.1, 3.4, 3.5.3, 5, 5, 5.3.2, 5.3.2, 5.4.1
- [97] Zhang, Z.; Guo, C.; and Goes, P. 2013. Product comparison networks for competitive analysis of online word-of-mouth. *ACM Trans. Manage. Inf. Syst.* 3(4). 2.4
- [98] Zhao, J.; Guan, Z.; and Sun, H. 2019. Riker: Mining rich keyword representations for interpretable product question answering. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, 1389–1398. 4
- [99] Zheng, L.; Noroozi, V.; and Yu, P. S. 2017. Joint deep modeling of users and items using reviews for recommendation. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, WSDM '17, 425–434. 4, 4.1
- [100] Zhuang, L.; Jing, F.; and Zhu, X.-Y. 2006. Movie review mining and summarization. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, 43–50. 2.3.4