Advanced Review Helpfulness Modeling

Thesis submitted in total fulfilment of the requirements for the degree of

Doctor of Philosophy

by

Jiahua Du

February 2020

ADVANCED REVIEW HELPFULNESS MODELING

Jiahua Du, PhD
Victoria University 2020

In recent years, online shopping has gained immense popularity due to its feedback mechanism. By composing online comments, previous buyers share opinions and experiences regarding the items that they have purchased. These user-generated reviews, in turn, provide valuable information to potential customers in regards to deciding which products to purchase. The reviews also help vendors understand customer needs and improve product quality. Yet despite these benefits, the unprecedentedly rapid growth of user-generated content has overwhelmed human ability in online review scrutiny. Online reviews that possess varying content further impedes useful knowledge distillation. The large volume of online reviews that are uneven in quality puts growing pressure on automatic approaches for effective review utilization and informative content prioritization.

Review helpfulness prediction leverages machine learning methods to identify and recommend helpful reviews to customers. In particular, review characteristics form the backbone of helpfulness information acquisition. Prior literature has observed and associated a large body of determinants with review helpfulness. However, these determinants heavily rely on the domain knowledge of experts. The selection of and the interaction between the determinants also remain understudied, leaving ample room for exploration. The general lack of systematic experiment protocols among the existing methods further harms the task's reproducibility, comparability, and generalizability.

This thesis aims to automatically model helpfulness information from online user-generated reviews. The thesis proposes effective modeling techniques and novel solutions to tackle the aforementioned challenges, with more emphasis on sophisticated feature learning and interaction. The thesis has made the following contributions to standardize the research field and advance the accuracy in helpfulness prediction.

1. A comprehensive survey is conducted to identify frequently used content-based determinants for automatic helpfulness prediction. A computational framework is developed to empirically evaluate the identified features across domains. Three selection scenarios are considered for feature behavior analysis. The domain-specific and domain-independent feature selection guidelines are summarized to

facilitate future research prototyping. The implementation details of the study are discussed to standardize the task of automatic helpfulness prediction.

2. A deep neural framework is designed to enrich the interaction between review texts and star ratings during automatic helpfulness prediction. A gated convolutional component is introduced to learns content representations. A gated embedding method is proposed for encoding sophisticated yet adaptive rating information. An element alignment mechanism is proposed to explicitly capture the text-rating interaction. Ablation studies and qualitative analysis are conducted to discover insights into the interactive behavior of star ratings.

3. An end-to-end neural architecture is proposed to contextualize automatic helpfulness prediction using review neighbors. Four weighting schemes are designed to encode a review's surrounding neighbors as its context information into content representation learning. Three types of reviews neighbors of varied length are considered during context construction. Finally, discussions on the experimental results and the trade-off between model complexity and performance are given, along with case studies, to understand the proposed architecture.

DOCTOR OF PHILOSOPHY DECLARATION

I, Jiahua Du, declare that the PhD thesis entitled "Advanced Review Helpfulness Modeling" is no more than 100,000 words in length including quotes and exclusive of tables, figures, appendices, bibliography, references and footnotes. This thesis contains no material that has been submitted previously, in whole or in part, for the award of any other academic degree or diploma. Except where otherwise indicated, this thesis is my own work.

Signature  Date 14/02/2020

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere gratitude and appreciation to my principle supervisor Professor Yanchun Zhang, my associate supervisor Professor Hua Wang, and my associate supervisor Doctor Jia (Jackie) Rong, for their professional guidance, continued encouragement, and constructive advice on my research. They also contributed valuable time in training my research writing and critical thinking skills, and provided abundant academic workshops throughout my PhD program. My thesis would not have been possible without the valuable help from these supportive supervisors. I would also like to thank their families, Doctor Jinli Cao, Doctor Lili Sun, PhD candidate Kate Wang, and Xinxue (Cindy) Zhang.

I would like to thank the chair of my review panel, Professor Randall W Robinson, and other panel members for their valuable time and comments on my work throughout the PhD candidature milestones. Professor Robinson has been considerably helpful and supportive to me and my research.

I would like to thank my colleagues listed alphabetically: Cheng Lee, Chuanchuan Zheng, Dinesh Pandey, Fan Liu, Feiyi (Aaron) Tang, Ferry Susanto, Geordie Zhang, Hui Zheng, Jiaying (Alice) Kou, Jinyuan (Sam) He, Le Sun, Luyao (Luna) Teng, Majid Afzalirad, Neda Afzaliseresht, Nithya Saiprasad, Peng (Mark) Zhang, Ravinder Singh, Roozbeh Zarei, Rubina Sarki, Saratkumar Rangarajan, Shekha Chenthara, Sudha Subramani, Supriya Angra, Xinyu (Pino) Cao, and Ye (Yesi) Wang. Thank you for the encouragement when I lost faith in myself and for sharing valuable research experiences. I would like to give special thanks to Sam and his family for their substantial help in my life; to Alice for her kindness, empathy, and open-minded thoughts which facilitated the development of my research topic; to Mark and Roozbeh for numerous research discussions and sharing life experience; to Yesi for offering help and useful suggestions to my research; to Ravinder for sharing his brilliant business insights. I would also like to thank all the visiting scholars and other members of the research group. I had delightful time working with all of you.

I appreciate Professor Yuan Miao, Associate Professor Hao Shi, Grace Tan, Gitesh Raikundalia, and Professor Jing He for offering me teaching opportunities at Victoria University, which have been a valuable experience.

I acknowledge China Scholarship Council for supplying the financial support, Victoria University for providing appropriate facilities, and AARNet for offering free online file sharing and storage services to ensure the viability of pursuing this doctoral degree.

I would like to show my respect to Professor Stephen Collins, Associate Professor

Dianne Hall, Doctor Lesley Birch, Elizabeth Smith, Parisima Nassirnia, Manira Osman, Joyce Nastasi, Meika Scholz, Martina Mariu, Khandakar Ahmed, Maria Arambulo, Stella Paraskevas, and other university staff for their assistance in university paperwork entailed over my PhD program.

I would like to thank Jiachun (Emma) Cai and Xianglin (Frank) Xu for their thoughtfulness and helpfulness. I wish them have a happy family with their baby girl Annie and Laughing world's number one handsome Siberian husky.

I would like to thank the lovely people in the research lounge: Bren Lovell, Eduardo Enrique Morales Vargas, Roopa Sreedhar, Robert Louis MacKenzie, Samuel King, Bassam Saleh, Thuy Dieu Linh Hoang, Bich Tien Ma, and Abdul Rahman, for together creating a positive, friendly, and reciprocal atmosphere for multi-disciplinary research. I will keep those wonderful memories in my heart forever.

Tremendous gratitude is owed to Sandra Michalska, who taught me resilience and imperturbability, for her elegance, inspiration, genuineness, and selfless assistance. It is my pleasure to have met her and shared her unique life philosophy and wonderful mind.

Particular appreciation is given to my life coach Douglas Pinto Sampaio Gomes, who enlightens my world, and in particular guides me towards strength training. He has been an extraordinary person that embodies perseverance, determination, sophistication, nearly inexhaustible knowledge, and astonishing versatility.

I am extremely grateful to Professor Gansen Zhao and Jiantao He, who have been showing generous willingness, patience, and care for me, treating me like a family; I am forever indebted to them. I would also like to thank my friends and former colleagues listed alphabetically: Aiping Li, Chengchuang Lin, Haiyu Wang, Haoxiang Tan, Xiaofeng Huang, Ziliu Li, Zilu Yuan, who helped me overcome challenges and become a better person.

In closing, I am forever grateful to my parents, grandparents, and other members of my extended family, for their unconditional love and understanding. I would like to specially thank my parents and also my uncle who have dedicated their time and effort in taking good care of my maternal grandmother. My grandmother is the most kind-hearted and considerate person I have ever met, who unfortunately suffers from chronic diseases that she does not deserve to. I pray for her good health and long life. I would also like to thank my beloved girlfriend for her supportive company on countless days and nights; for bringing me joys and hopes; for sharing ups and downs; for always believing in me. I love you all.

PUBLICATIONS

The following research articles have been published in or submitted to international journals and conferences.

- Jiahua Du, Jia Rong, Hua Wang, and Yanchun Zhang. Helpfulness prediction for online reviews with explicit content-rating interaction. In *Web Information Systems Engineering (WISE)*, pages 795–809, Hong Kong SAR, China, October 2019. Springer International Publishing
- Jiahua Du, Jia Rong, Sandra Michalska, Hua Wang, and Yanchun Zhang. Feature selection for helpfulness prediction of online product reviews: An empirical study. *PLOS ONE*, 14(12):1–26, December 2019
- Jiahua Du, Sandra Michalska, Sudha Subramani, Hua Wang, and Yanchun Zhang. Neural attention with character embeddings for hay fever detection from twitter. *Health Information Science and Systems*, 7(1):21, October 2019
- Jiahua Du, Liping Zheng, Jiantao He, Jia Rong, Hua Wang, and Yanchun Zhang. An interactive network for end-to-end review helpfulness modeling. *Data Science and Engineering*, pages 1–19, 2020
- Jiahua Du, Jia Rong, Hua Wang, and Yanchun Zhang. Exploiting review neighbors for contextualized helpfulness prediction. Submitted to Decision Support Systems
- Jiaying Kou, Xiaoming Fu, Jiahua Du, Hua Wang, and Geordie Z Zhang. Understanding housing market behaviour from a microscopic perspective. In *2018 27th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–9. IEEE, July 2018
- Sudha Subramani, Sandra Michalska, Hua Wang, Jiahua Du, Yanchun Zhang, and Haroon Shakeel. Deep learning for multi-class identification from domestic violence online posts. *IEEE Access*, 7:46210–46224, April 2019
- Jia Rong, Sandra Michalska, Sudha Subramani, Jiahua Du, and Hua Wang. Deep learning for pollen allergy surveillance from twitter in Australia. *BMC medical informatics and decision making*, 19(1):208, November 2019
- Shanshan Qi, Cora Un In Wong, Ning Chen, Jia Rong, and Jiahua Du. Profiling Macau cultural tourists by using user-generated content from online social media. *Information Technology & Tourism*, 20(1):217–236, December 2018

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

The advent of e-commerce has dramatically changed a wide range of shopping activities. Nowadays, people participate online shopping to share and enjoy the convenience brought by the new business paradigm. From ordering food in restaurants to booking hotel rooms, more and more transactions are completed online or even via mobile environments. One attractive advantage of e-commerce is the access to comprehensive product information about goods and services. Contemporary online shopping platforms often solicit feedback from previous customers, predominantly in the form of product reviews. The reviews share crowd-sourced opinions, feelings, and experience towards products, from which potential customers can seek advice to make more informed purchase decisions. Additionally, manufacturers and retailers can learn from the collective wisdom to understand customer needs and facilitate future product development.

Currently, e-commerce platforms have accumulated a plethora of user-generated reviews. The accumulation speed is also increasing. In fact, existing online reviews have exceeded the capability of human scrutiny within acceptable time limits. This phenomenon poses new challenges to both individuals and companies in decision and strategy making. The review ranking mechanism via human-involved quality evaluation has long been adopted in many online platforms to combat information overload. Still, the methods are primitive and far from effective in terms of review utilization. The large scale and rapid growth of online reviews require for informative content prioritization. As a result, efficacious solutions are required to filter low-quality content and locate useful information, in an automatic manner.

1.1 Background

This section further introduces background knowledge regarding automatic helpfulness prediction. Specifically, three dimensions will be discussed: the rapid growth of e-commerce and its significant influence; the importance of online user-generated reviews and information overload hindering review exploration and utilization; and current review ranking mechanisms that help customers to read reviews in a more effective manner.

1.1.1 Electronic Commerce

As recently as the mid-1990s, e-commerce was still in its infancy. With the help of Web 2.0 and Internet technologies, e-commerce has developed into a worldwide industry [272] worth 2.9 trillion US dollars. E-commerce is one of the most important online activities. According to a recent survey conducted by Episerver [82], 26% (62%) of customers shop online on a weekly (monthly) basis in 2019; nearly a quarter of the online buyers access e-commerce environments daily, and nearly half do so multiple times per week. The influence of e-commerce continues worldwide. Figure 1.1 exemplifies the trend of e-commerce in US and UK using the national sales data. As illustrated, both countries receive an overall increase in proportion of online transactions in over the past decade. By 2021, 2.14 billion people [285] are expected to buy goods and services online. By the end of 2040, e-commerce is thought to facilitate 95% of purchases [147]. The global e-commerce sales are predicted to hit 6.5 trillion US dollars [169] in 2023, reaching 22% of total retail sales.

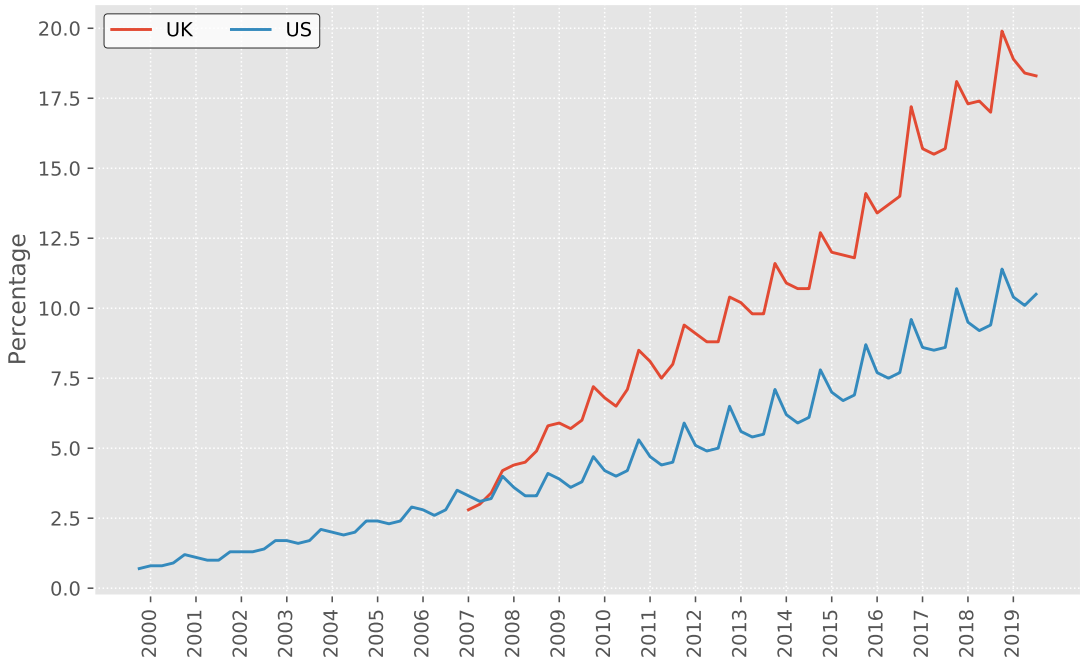

Figure 1.1: Estimated quarterly e-commerce sales as a percent of total sales. Sourced from the US Census Bureau and Office for National Statistics.

E-commerce has infiltrated into people's daily lives, in areas ranging from restaurant reservations, hotel booking, product purchasing, to different kinds of appointments.

Even offline retailers can benefit from e-commerce. For example, Walmart managed to obtain double-digit e-commerce growth [89] in the fiscal years 2017, 2018, and 2019. In recent years, the number of e-commerce related businesses and applications has been rapidly growing. In 2019, North America was estimated to have 1.3 million e-commerce companies [218]. As of 2020, nearly 12 million live websites over the entire Internet [34] are observed using e-commerce technologies to improve business quality and experience. The appeal of e-commerce over traditional in-store shopping [62] lies in the spatiotemporal convenience, which enables customers to shop at any time without physically entering stores. Additionally, e-commerce platforms allows customers to compare prices of a wide range of items and select better price options in one place. More importantly, as will be discussed in the next subsection, such platforms are equipped with user-generated online reviews that provide customers with rich information for knowledge learning and decision making.

1.1.2 User-generated Reviews

Online reviews have become integral components of contemporary e-commerce ecosystems. Currently, user-generated reviews are building blocks of many web communities. In 2018, online reviews were believed to influence 15.44% Google search result rankings [273], up from 10.8% in 2015. In the context of e-commerce, the influence is even more pronounced.

Online reviews as information acquisition methods have become common practice. A recent survey conducted by Bizrate Insights [139] shows that approximately 98% of online shoppers conduct research on a vendor via online reviews. Among the respondents, 24.4% report that they always prefer such a paradigm, while 40.8% report that they often prefer such a paradigm. Similar statistics can also be found in the tourism domain [1], where 95% of travelers read reviews prior to booking, with 59% always or very often doing so. Online reviews are considered to play crucial roles in decision making. A Fan & Fuel's survey [95] reveals that 97% of participators believe that online reviews factor into their buying decisions. A survey conducted by Capterra [118] has found that online reviews impact the purchase decisions of almost all software buyers. 85% of US Internet users [215] trust online reviews as much as recommendations from personal sources such as friends and family; the number becomes 91% [216] for those aged between 18 and 34. Online reviews are also indispensable to providers of

goods and services. From the vendors' perspective, online reviews can be researched to promote product quality, analyze user satisfaction [2], and explore user needs.

Online comments provide a reliable source of reference [140] that improves customers' confidence, comfort, and experience. During purchasing, only 34% of buyers [252] consider vendor-related content to be trustworthy. Instead, 66% of customers [167] use sources outside of vendor materials during the research phase. Nearly two-thirds of US customers [296] agree that reviews created by online users are more interesting than brand-generated content. For mothers who use the Internet, the reviews can be trusted almost 12 times [81] more than the manufacturers-provided descriptions. As pointed out in a recent social influence study [64], online peer reviews are not only more trusted by 68% customers; they are also 16% more memorable than traditional media. The fascination with online reviews probably relates to an awareness of the pros, cons, and user experiences of products [95] from a variety of customers. For example, 52% of consumers [118] believed that a software product that has received negative reviews will be more trustworthy. Apart from unilateral manufacturer-provided information, customers can now rely on crowd-sourced opinions to make more informed purchasing decisions.

Despite the advantages described above, customers are facing new challenges in efficiently exploiting online reviews. Since 2008, there has been a boom in online customer reviews. An example is the platform Yelp. As Figure 1.2 shows, the cumulative number of customer reviews posted on the platform has risen from $4,689$ in 2008 to $177,385$ in 2018, increasing nearly 37 times in ten years; the yearly growth is still accelerating. Even for niche items, the number of reviews can typically exceed 200 [226]. Such volume has exceeded manual power to digest all reviews a product can receive. In addition, online reviews are of uneven quality. The written content of reviews depends on customers' life experience, education background, and purposes for writing reviews. While some reviews are informative, others provide little value. Consequently, customers may require additional time and effort to read reviews in order to gather sufficient information to make purchase decisions. Moreover, customers have been shown to have limited patience for perusing review. Most customers read less than 10 reviews before forming an opinion [214] or even decision [13] about a business/product. For example, it takes on average less than seven reviews [1] for travelers to make hotel booking decisions. In the first quarter of 2019, the average time consumers spent in online stores [262] dropped to 4 minutes and 12 seconds. The large volume and unpredictable quality of reviews, along

with limited customer patience, demands better review utilization strategies to manage the information overload.

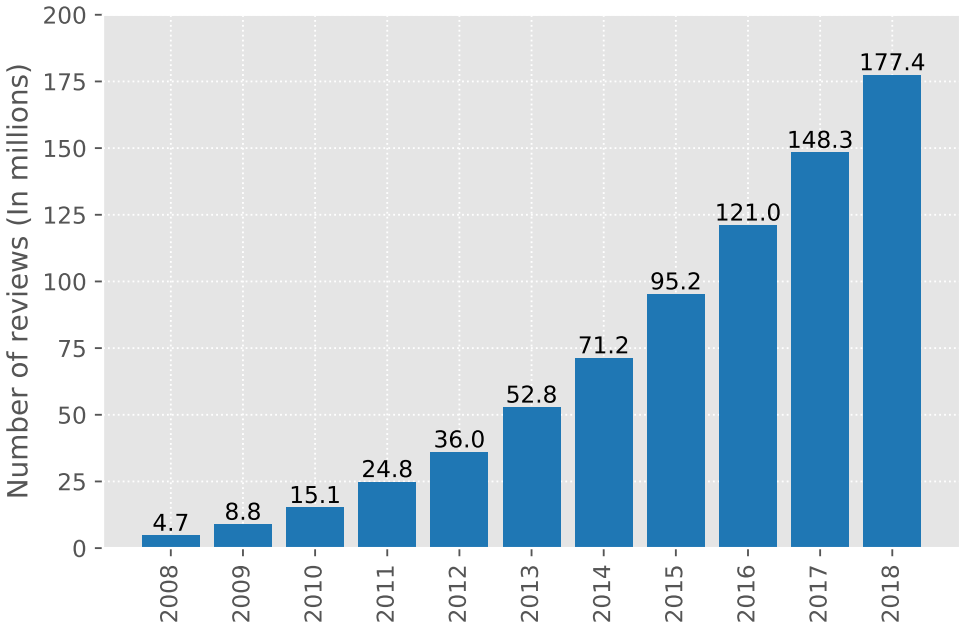

Figure 1.2: The cumulative number of reviews submitted to Yelp from 2008 to 2018. Sourced from US Securities and Exchange Commission.

1.1.3 Review Ranking

Contemporary online shopping platforms have taken several measures to enable users to read review more efficiently. One standard method is to solicit feedback from customers towards other customers' opinions in addition to feedback towards products. By asking "Was this review helpful to you?", or "Did you find this review helpful?" at the end of each review, online platforms can crowdsource dichotomous helpfulness votes from customers. As a result, the collected voting data of individual reviews are shown to reflect how people generally think of the review helpfulness. Figure 1.3 exemplifies the feedback mechanism on Amazon, to which that on other platforms are similar. As user feedback accumulates, the received votes reflect how consumers generally think of the helpfulness of these reviews. Thus, the voting data can be used to measure review quality and rank reviews by their quality, making the platform a self-managing system. In fact, questions such as those described above are responsible for more than 2.7 billion

US dollars of new revenue[1] for Amazon every year.

Figure 1.3: Helpfulness voting examples. Sourced from Amazon Web Services documentation.

Despite the advantages, current voting mechanisms can be problematic. First, the voting data suffers from scarcity [277] since only a small proportion of customers [120] in the community are willing to vote for review helpfulness. As Figure 1.4shows, the voting numbers follow a power-law distribution. The scarcity is even more severe in reviews of low-traffic (less popular) products [85] and recently submitted reviews [175]. Moreover, the voting data suffers from unexpected biases. Online platforms often dynamically rank reviews using helpfulness-related voting algorithms. The ranking, for example, can simply fall into the winner-take-all bias [172]. That is, more votes help reviews to gain higher rankings, which in turn is more likely to attract for helpfulness votes. Both situations limit review ranking to a small range of reviews, whereas the remaining valuable reviews are ignored. As such, some potentially helpful reviews [38] are unnecessarily ranked as "unhelpful". Additionally, the voting system is vulnerable to spam reviews [131, 328] and can be abused for the purpose of voting manipulation. The aforementioned drawbacks will diminish the reliability of the obtained votes.

Given the scenarios described above, a substituted methodology for human helpfulness assessment that predicts the helpfulness of evaluative reviews in an automatic

[1] https://articles.uie.com/magicbehindamazon/

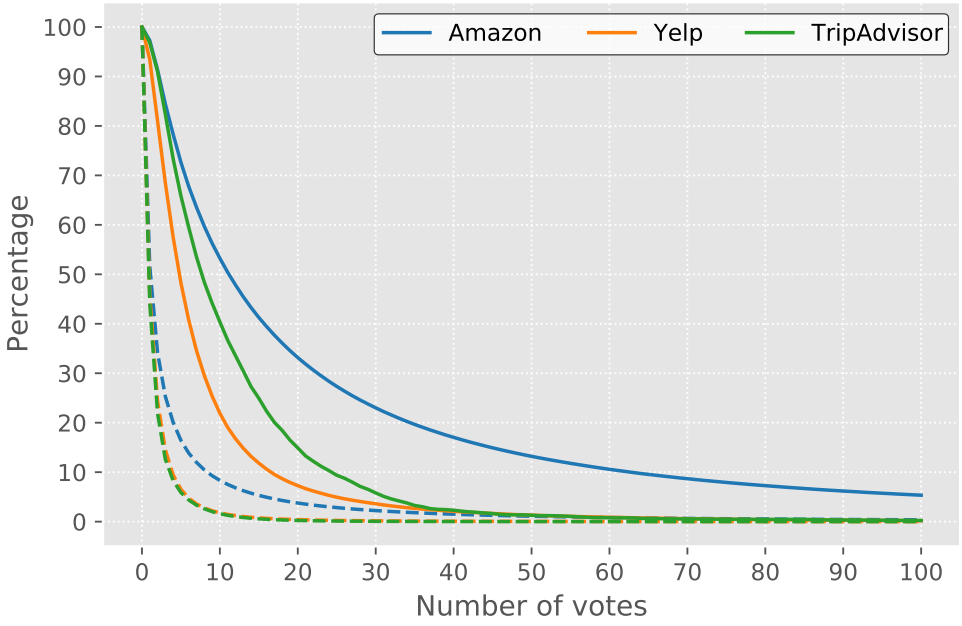

Figure 1.4: The scarcity of voting data in reality. Solid (Dotted) lines indicate the percentage of remaining reviews (products with reviews) that have at least a certain number of votes. Data retrieved from three popular online platforms: Amazon [113], Yelp [324], and TripAdvisor [164].

manner will be desirable. Automatic helpfulness prediction [138] aims to better utilize the collected voting data to identify and recommend high-quality reviews to customers. This interdisciplinary field of research involves psychology, sociology, information technology, human-computer interaction, human behavior analysis, and marketing. The goal of automatic helpfulness prediction is to adopt machine learning techniques to approximate the helpfulness assessment process using knowledge gained from previously voted reviews.

1.2 Research Motivations and Problems

This thesis aims to provide methods of automatic helpfulness prediction. The ultimate goal is to improve the identification accuracy of high-quality reviews. To this end, data-driven models are constructed to develop effective review representations for helpfulness information modeling. The following problems will be specifically discussed.

1. **How to identify and select robust features for generalized helpfulness prediction?**

   The mainstream approach of automatic helpfulness prediction [254] focuses on feature engineering. This paradigm has several issues that are yet to be resolved. The curation and organization of feature candidates tends to be arbitrary and lack justification. For example, the use of one feature in preference to other similar features is often ignored. Certain features are also frequently domain- and/or platform-dependent. Moreover, the curated features are often evaluated as a whole, whereas the predictive power of feature subsets is largely understudied. Without effective selection strategies, overusing the features may cause redundancy that harms helpfulness modeling. Furthermore, many existing works are barely comparable and reproducible due to unclear experiment details [107] and unavailable ad-hoc datasets [224]. Consequently, systematic empirical analysis of feature identification and selection is required to ensure result and finding generalizability.

2. **How to learn interactive review representations from review texts and star ratings?**

   Text content and star ratings are noticeable elements that attract readers' attention when they are perusing reviews. The two factors are arguably important for helpfulness conception: text content qualitatively describes reviewers' opinions, whereas star ratings express their quantitative attitudes towards an item under review. More importantly, the (in)consistency between the two factors can affect a consumer's ability to trust a review. For instance, a positive review with a negative rating may cause confusion to and thus become less helpful for consumers. Currently, review texts have been widely exploited for learning helpfulness-related features. The exploitation of star ratings, however, remains understudied in terms of the encoding of rating information and the interaction with the content of a text. To overcome the limitations and better use of rating information, more sophisticated approaches are proposed to increase the encoding capability of star ratings and to strengthen the interactive power between review texts and star ratings.

3. **How to learn contextualized review representations from the surrounding context?**

   The evaluation of review quality relies strongly on the historical helpfulness voting data provided by customers. The vast majority of existing studies assume that customers independently vote on whether a review is helpful: the helpfulness of

a review only depends on the review's content and will not be affected by other reviews surrounding it. In reality, online reviews are displayed in the form of sequences. During review perusal, different reading orders may lead to customers perceiving review helpfulness in different ways. The reason is that the opinions from past reviews read by a customer affects his/her thoughts on the current review and future ones. As such, the helpfulness of a review is closely related to how and where it is presented, its surrounding context. In this thesis, contextualized models are proposed to extend the self-contained assumption, taking into account the surrounding context of a review in helpfulness modeling.

1.3 Thesis Contributions

This thesis contributes to existing literature and research community from at least four perspectives:

1. An empirical study on feature identification, selection, and evaluation for automatic helpfulness prediction on online product reviews.
    - A comprehensive literature surveys on helpfulness-related determinants and the identification of frequently used content-based features.
    - A flexible model for feature extraction and extensive empirical validation on large-scale publicly available online product reviews from a variety of domains.
    - Quantitative evaluation on feature behaviors in multiple feature selection scenarios (individual features, features within the same category, and the whole feature collection) across domains.
    - Qualitative analysis of the trained prediction models and investigation into the effectiveness of features with case studies.
    - The release of the dataset split configurations, pre-processed reviews, and extracted features for result reproducibility, benchmark studies, and further improvement.
2. A deep interactive neural network considering both review texts and the accompanying star ratings for automatic review helpfulness prediction.

- A gated convolutional encoder that learns content representations from review texts by capturing deep semantic relationships between words.
- An embedding method for review star ratings to enlarge feature space for rating information encoding.
- An explicit interaction mechanism between review texts and star ratings via element alignment between the learned content representations and rating embeddings.
- An adaptive learning method that specifies the amount of rating information needed for the learned content representations during interaction.
- Quantitative benchmarking against robust baselines, ablation studies into model effectiveness, and qualitative analysis of the learned rating embeddings and their behaviors with case studies.

3. An end-to-end neural architecture that takes into account the surrounding context of a review for automatic review helpfulness prediction.

    - A novel method in using a review's neighbors as its surrounding context into helpfulness modeling.
    - A novel dataset created from scratch to fulfil the task. The dataset consists of six domains of reviews collected from two real-world online platforms. Each review contains up to ten preceding and following neighbors.
    - Extensive evaluation of a set of settings designed for constructing contextual information from a number of review neighbors, including three neighbor types (i.e. preceding, following, and surrounding reviews) and four weighting schemes.
    - Ablation analysis and hyperparameter tuning to locate the optimal context settings and to discuss the trade-off between model complexity and effectiveness.
    - Qualitative analysis via visualization and case studies for better understanding and model interpretation.

4. Some of the results of this thesis will be further organized and integrated into an ongoing project called **Help**fulness **Me**asurement (HelpMe). This project consists of two components:

- An all-in-one computational framework for generic helpfulness prediction tasks. The framework consists of a series of built-in functions (e.g., review pre-processing, review labeling, review splitting, feature extraction, and embedding training) that facilitate data processing, model prototyping, and baseline evaluation. At present, the framework has integrated the models proposed in this thesis and will continue to include more published helpfulness-related features and prediction models.
- A summarized table from state-of-the-art helpfulness prediction studies for quick reference. The table contains brief summaries of a series of up-to-date high-quality publications related to the topic. Each summary focuses on a paper's methodology (e.g. features and classification/regression models) and experiment details (e.g. datasets and evaluation metrics). Meanwhile, more published helpfulness-related features and prediction models will be added to the table.

The release of the computational framework fills the gap within existing studies, which lack systematic experiment standards and reproducible implementations. Future researchers are strongly encouraged to adopt the framework for data preparation and model development to achieve a higher level of result comparability and reproducibility. The release of the summarized table is also beneficial for current and future researches because it offers a fast and comprehensive understanding of the past and present trends in relation to automatic helpfulness prediction.

1.4 Thesis Structure

The remainder of the thesis is organized as follows:

Chapter 2 conducts a comprehensive literature review on the task of automatic helpfulness prediction to report existing definitions, determinants, methodologies, and data sources for modeling helpfulness information from online user-generated reviews. The chapter also provides an overview of common text representation techniques.

Chapter 3 conducts a systematic study to identify frequently employed context-based features for automatic helpfulness prediction. Empirical evaluation is then con-

ducted on the identified features to locate optimal feature combinations for the task under multiple feature selection scenarios, followed by the summary of domain-specific and domain-independent feature selection guidelines.

Chapter 4 presents a deep neural architecture to capture the complicated interaction between review texts and star ratings when predicting review helpfulness. The model is benchmarked against state-of-the-art solutions on real-world online reviews, along with ablation studies and detailed qualitative analysis.

Chapter 5 describes a deep end-to-end neural architecture that contextualizes a review in relation to the reviews that surround it during the helpfulness learning process. The framework is evaluated on real-world online reviews, along with ablation studies, parameter tuning, and detailed qualitative analysis.

Chapter 6 concludes the thesis by summarizing the findings and practical implications gleaned from the experimental results. Potential future directions are enumerated to encode more accurate review representations that further improve the helpfulness prediction performance.

CHAPTER 2

LITERATURE REVIEW

This chapter surveys literature on automatic review helpfulness prediction. The survey first overviews common review text representation techniques, followed by three fundamental perspectives of helpfulness prediction: the helpfulness voting process on contemporary online platforms, the interpretation of review helpfulness, and factors that online users perceive reviews as helpful/unhelpful. Subsequently, the survey discusses frequently-employed determinants modeling helpfulness cues, introduces methods that label review helpfulness, and presents review sources for the task

2.1 Text Representation Overview

Representing texts is of vital significance to many real-world applications, including review helpfulness prediction. In accordance with Natural Language Processing (NLP) and information retrieval conventions, a text (e.g., a sentence, paragraph, or document) is encoded in the form of vectors. This section categorizes text representations into local and continuous ones [200] and briefly introduces existing techniques used to learn both types of representations.

2.1.1 Local Representations

Local encoding focuses on the one-to-one correspondence between physical entities (e.g., characters, words, tokens) of a text and computing elements. The one-hot encoding scheme, also known as the 1-of-N encoding scheme, is a standard method for word representation. Given a collection of texts, the scheme first constructs and indexes the vocabulary of unique tokens in the corpus. Each token is represented by a sparse vector of the same length as the vocabulary, The vector encodes one into the element indicating the token's position and zeros otherwise.

Bag-of-Words (BOW) models represent a text by aggregating the one-hot vectors of its constituent tokens. Three scoring schemes and their variants are frequent used: binary values indicating token presence/absence in a text, integers counting token occurrences, and the standard Term Frequency Inverse Document Frequency (TFIDF) scheme [264] being the most popular option. TFIDF states that the importance of a token relies

on the token's occurrence in a text and the number texts in the corpus containing the token.

BOW models by definition disregard word orders and thus cannot distinguish between texts consisting identical but differently arranged words, for example, "is it true" and "it is true". To alleviate the drawback, n-gram models take as input spatial adjacency by encoding contiguous sequences of n constituent tokens of a text. Frequent n-gram options include individual tokens ($n = 1$), token pairs ($n = 2$), and token triplets ($n = 3$), also known as unigrams, bigrams, and trigrams, respectively. Note that a BOW models can be seen as a 1-gram model.

Still, n-gram models may suffer from the curse of dimensionality [201]. A common corpus usually entails a vocabulary of 10^5–10^7 tokens and the vocabulary size will grow exponentially if using higher-level n-grams. Besides computational inefficiency, n-gram models pose large sparsity as many n-grams only have few occurrences and carry little semantic information. Since vector elements are treated independently, n-gram models cannot capture synonyms (e.g., "lemon juice" and "lemonade") and hypernyms (e.g., "husky" and "dog") and other semantic relationships.

2.1.2 Continuous Representations

Continuous encoding uses more complicated implementations by associating a physical entity with a series of shared computing elements. Topic modeling, which is a branch of the family, assumes that a text is governed by a mixture of hidden topics and each topic a collection of terms in the corpus. To reduce sparsity while preserving most of the semantic meaning, a topic model decomposes the document-term matrix resulting from BOW or n-gram representations into a document-topic matrix and a topic-term matrix. The compressed vector space encodes more abstract semantics such as topical and aspectual information. Two classical topic modeling techniques are Latent Semantic Analysis (LSA) [70] and Latent Dirichlet Allocation (LDA) [30].

Distributed representations mark another branch of the continuous family. As a consequence of training neural language models [23], each token is mapped into a fixed-length real vector (i.e., embedding), wherein each dimension represents a latent concept shared across tokens. In contrast to local representations, an embedding comprises 10^1 – 10^3 computing elements and thus is immune to dimensional

disaster. The embedding training process takes into account the a word's local context to capture more sophisticated semantic relationship among words. The intuition is inspired by distributional hypothesis [92, 111] in linguistics: words that occur in the same contexts tend to have similar meanings. As a result, similar words in meaning are spatially closer in the trained vector space. The learned representations also entail word analogies [202] via simple algebraic operations. For example, vector("King")−vector("Men")+vector("Women")≈vector("Queen").

Early embedding training methods harvest the success of shallow neural networks for word semantics learning. Three classical techniques [201, 237] for learning dense word vectors are the Continuous Bag-of-Words (CBOW) model, Skip-Gram model with Negative Sampling (SGNS), and Global Vectors (GloVe). The learning paradigm can also be applied to subwords [133] and other language units. The representation of a text (or a document) [10] is obtained by (i) the (un)weighted average of the trained vectors of its constituent tokens [274], (ii) learning along with the token vectors [157, 66], or (iii) developing another neural model upon the token vectors [128].

Below describes neural architectures for learning the compositionality of document embeddings. Due to limited space, this chapter only discusses techniques used in existing helpfulness prediction studies. The mathematical notation for model description is denoted as follows. Given a text $s = (w_1, w_2, \ldots, w_T)$ of T words, each word w is mapped into a d-dimensional word embedding x, and thus the text can be represented as a matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T] \in \mathbb{R}^{T \times d}$ of stacked embeddings. The goal of the networks is to learn the document embedding $\mathbf{s}$ of s from $\mathbf{X}$ through different types of non-linear matrix transformations.

- **Convolution Neural Networks** (CNNs) [142] encode the document embedding of s from its n-gram information. Let $\mathbf{c}_t \in \mathbb{R}^{nd}$ be the concatenation of the n-gram embeddings $\mathbf{x}_{t-n+1}, \mathbf{x}_{t-n+2}, \ldots, \mathbf{x}_t$, where $0 < t < T + n$. An embedding $\mathbf{x}_t$ is set as zero vectors when $t < 0$ and $t > n$. A vanilla CNN framework employs a set of K learnable kernels $\mathbf{W}_c \in \mathbb{R}^{K \times nd}$ of the same filter size n to compute the convolution between a kernel $\mathbf{w}_c \in \mathbb{R}^{nd}$ and $\mathbf{c}_t$. Each kernel slides over the n-gram embeddings of $\mathbf{X}$, resulting in T convoluted features, also known as feature maps. The K kernels together produce $\mathbf{P} \in \mathbb{R}^{K \times T}$:

$$\mathbf{P}_{*,t} = \tanh(\mathbf{W}_c \circledast \mathbf{c}_t + \mathbf{b}), \tag{2.1}$$

where $\circledast$ indicates the convolution operation and $\tanh(\cdot)$ the hyperbolic tangent function. The text representation $\mathbf{s}$ is then computed by concatenating salient n-gram features in each of the K kernels.

$$\mathbf{s} = \bigoplus_{k=1}^{K} \max(\mathbf{P}_{k,*}), \tag{2.2}$$

where $\oplus$ is the concatenation operation and $\max(\cdot)$ the max-over-time pooling operation that only preserves the top-m values in a feature map, with $m = 1$ used frequently. In practice, multiple n values can be employed simultaneously to capture salient features from different text n-grams.

- **Recurrent Neural Networks** (RNNs) [170] learn the representation of s by exhibiting the temporal dynamics of the word sequence. A vanilla RNN framework maintains an internal state $\mathbf{h}$ (also known as memory) to memorize the information of words that have been processed at a certain time. Given a time step $t \in [1, T]$, the framework takes as input both the word embedding $\mathbf{x}_t$ and the hidden state $\mathbf{h}_{t-1}$ learned from the previous step $t - 1$ to produce the hidden state $\mathbf{h}_t$. The current state $\mathbf{h}_t$ again is copied as the context of the processed words $w_1, w_2, \ldots, w_t$ for learning the hidden state at the next time step $t + 1$:

$$\mathbf{h}_t = \sigma(\mathbf{W}_h \mathbf{x}_t + \mathbf{U}_h \mathbf{h}_{t-1} + \mathbf{b}_h), \tag{2.3}$$

  where $\mathbf{W}_h$, $\mathbf{U}_h$, and $\mathbf{b}_h$ are weights and biases to be estimated during training. As a result, the framework yields T outputs $[\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_T]$ respectively for the individual entities in s. Often, the document embedding $\mathbf{s} = \mathbf{h}_T$ refers to the hidden state at the last time step that contain all information of the text.

  Despite that RNNs by nature are advantageous for handling sequential data, the vanilla RNN framework can suffer from gradient vanishing and exploding problems [24]. As such, training original RNN models on longer sequences is practically challenging. Two remedies modify the representation learning process using gating mechanisms.

  - **Long Short-term Memory Networks** (LSTMs) [116] augment RNNs with three types of gates controlling the flow of the inputs and outputs of the architecture. During text encoding, a cell state is maintained in through model training to adaptively add new information regarding a current token about and remove learned knowledge from the past inputs, allowing for learning

long-term dependencies. A vanilla LSTM block is formally defined as follows:

$$\mathbf{f}_t = \sigma(\mathbf{W}_f \mathbf{x}_t + \mathbf{U}_f \mathbf{h}_{t-1} + \mathbf{b}_f), \tag{2.4}$$

$$\mathbf{i}_t = \sigma(\mathbf{W}_i \mathbf{x}_t + \mathbf{U}_i \mathbf{h}_{t-1} + \mathbf{b}_i), \tag{2.5}$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_o \mathbf{x}_t + \mathbf{U}_o \mathbf{h}_{t-1} + \mathbf{b}_o), \tag{2.6}$$

$$\tilde{\mathbf{c}}_t = \tanh(\mathbf{W}_c \mathbf{x}_t + \mathbf{U}_c \mathbf{h}_{t-t} + \mathbf{b}_c), \tag{2.7}$$

$$\mathbf{c}_t = \mathbf{f}_t \otimes \mathbf{c}_{t-1} + \mathbf{i}_t \otimes \tilde{\mathbf{c}}, \tag{2.8}$$

$$\mathbf{h}_t = \mathbf{o}_t \otimes \sigma(\mathbf{c}_t), \tag{2.9}$$

where $\otimes$ is the Hadamard (i.e., element-wise) product and $\sigma(\cdot)$ the sigmoid function. The forget gate $\mathbf{f}_t$, the input gate $\mathbf{i}_t$, and the output gate $\mathbf{o}_t$ jointly influence the cell state $\mathbf{c}_t$, which is used to produce the output $\mathbf{h}_t$ at time step t. The weights and biases $\{\mathbf{W}_f, \mathbf{U}_f, \mathbf{b}_f, \mathbf{W}_i, \mathbf{U}_i, \mathbf{b}_i, \mathbf{W}_o, \mathbf{U}_o, \mathbf{b}_o, \mathbf{W}_c, \mathbf{U}_c, \mathbf{b}_c\}$ are learnable parameters. Similarly, the output at the last time step $\mathbf{h}_T$ is often selected as the document embedding $\mathbf{s}$.

– **Gated Recurrent Units** (GRUs) [51] share a similar idea of LSTMs. Two types of gates are introduced to control the information flow in the architecture: the reset gate $\mathbf{r}_t$ determines the combination between a new input and previous memory and the update gate $\mathbf{z}_t$ specifies the amout of previous memory to be kept. A vanilla GRU block is formally defined as follows:

$$\mathbf{z}_t = \sigma(\mathbf{W}_z \mathbf{x}_t + \mathbf{U}_z \mathbf{h}_{t-1} + \mathbf{b}_z), \tag{2.10}$$

$$\mathbf{r}_t = \sigma(\mathbf{W}_r \mathbf{x}_t + \mathbf{U}_r \mathbf{h}_{t-1} + \mathbf{b}_r), \tag{2.11}$$

$$\tilde{\mathbf{h}}_t = \sigma(\mathbf{W}_h \mathbf{x}_t + \mathbf{U}_h (r_t \otimes \mathbf{h}_{t-1}) + \mathbf{b}_h), \tag{2.12}$$

$$\mathbf{h}_t = \mathbf{z}_t \otimes \mathbf{h}_{t-1} + (1 - \mathbf{z}_t) \otimes \tanh(\tilde{\mathbf{h}}_t), \tag{2.13}$$

The weights and biases $\{\mathbf{W}_z, \mathbf{U}_z, \mathbf{b}_z, \mathbf{W}_r, \mathbf{U}_r, \mathbf{b}_r, \mathbf{W}_h, \mathbf{U}_h, \mathbf{b}_h\}$ are learnable parameters. The architecture mainly differs from LSTMs in that (i) the forget and input gate are coupled by the update gate, (ii) the cell state is no longer maintained but instead merged into the exposed hidden state, and thus (iii) the output gate is no longer needed. As a result, a vallina GRU block is less complex and more computationally efficient than the LSTM counterpart.

In practice, bidirectional RNNs [271] are frequently used to enable the awareness of both past and future information about the text at every time step. The backward

state $\overrightarrow{\mathbf{h}_t}$ encodes from w_1 to w_T, whereas the forward state $\overleftarrow{\mathbf{h}_t}$ encodes the opposite direction.

- **Attention Mechanisms** [17] have been used in many tasks to improve vector representation learning. In RNNs, the last hidden state $\mathbf{h}_T$ is expected to summarize the entire word sequence s and thus often selected as the final text representation. Those document embeddings, however, have difficulties in remembering earlier text information especially in longer sequences. Attention mechanisms intend to construct document embeddings using all the intermediate hidden states $[\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_T]$ during sequence processing. A general form of attention employs a context vector to measure the importance of each hidden state:

$$\mathbf{u}_t = \tanh(\mathbf{W}_u \mathbf{h}_t + \mathbf{b}_u), \tag{2.14}$$

$$\alpha_t = \frac{\exp(\mathbf{u}_t, \mathbf{u}_q)}{\sum_{t=1}^{T} \exp(\mathbf{u}_t, \mathbf{u}_q)}, \tag{2.15}$$

$$\mathbf{s} = \sum_{t=1}^{T} \alpha_t \mathbf{u}_t, \tag{2.16}$$

  where $\mathbf{W}_u$, $\mathbf{b}_u$, and $\mathbf{u}_q$ are learnable parameters. The context vector $\mathbf{u}_q$ can be thought of as the correlation between individual words and a text in building the document embedding $\mathbf{s}$ for a given task. The weights α are normalized through softmax operations into a probability distribution. The representation $\mathbf{s}$ is then computed as the weighted sum of the hidden states.

Recent studies have used a series of variants of the aforementioned techniques in many NLP tasks, such as character embeddings to handle out-of-vocabulary words by capturing subword information, hierarchical implementations [323] to learn fined grained semantic compositionality, and the combination of CNNs and RNNs [156, 344] to utilize the local and temporal strengths of both network structures. More recently, pre-trained language models [239, 71, 338] for text representation also gains tremendous attention. Further discussions of such is outside the scope of the thesis. For sensitivity analysis of and systematic comparison between the techniques, one can refer to recent studies [336, 134, 327] on the topic.

2.2 Human Helpfulness Assessment

This section digs into the feedback mechanisms behind human helpfulness assessment. Understanding how readers perceive review helpfulness can facilitate automatic helpfulness modeling. The following subsections respectively discuss three core questions: (1) How do users rate review helpfulness? (2) What do users think helpfulness is about? and (3) What do users think helpfulness depend on? Here, reviews can be derived from a product, restaurant, attraction, hotel, type of service, to name a few.

2.2.1 Helpfulness Voting Process

Contemporary online platforms both collecting user-generated opinions and crowdsource the helpfulness of the opinions. Helpfulness voting can be explained using message and information processing theories [108]. Specifically, the voting process goes through a series of stages, from exposure (the presence of reviews), reception (the attention to and comprehension of reviews), to yielding (the evaluation, belief change, and attitude change towards review helpfulness). Ocampo et al. [224] define helpfulness voting as a three-step process: (1) a reviewer writes a review on a product; (2) a rater reads and assigns an internal score to the review based on certain criteria; and (3) if the score exceeds some threshold, the rater votes the review as helpful and unhelpful otherwise.

Rating review helpfulness can be fundamentally different from rating an item, mainly because the former is intertwined with complex social involvement. Message evaluation [153] depends not just on the information one pays attention to, but also his/her comprehension that reaches the evaluation. For example, a reader's needs [224] are unobservable unless explicitly informed. A review's perceived helpfulness also depends on both itself and its interaction [68, 280] with other reviews. For online environments where social functions are enriched [291], both raters (reviewers) and their interactions can provide social context about reviews that influences helpfulness voting.

Most online platforms dichotomize review helpfulness [7] so that customers can choose to vote yes (no) to a review if they think the review is helpful (unhelpful), or choose not to vote if the review is neither helpful nor unhelpful to them. The last option is not discussed since the total number of users who read reviews without voting is

unknown (at least by outsiders). Currently, an increasing number of online platforms only accept votes when users find a review helpful. Such voting mechanism alleviates voting abuse and manipulation [250, 241] and helps cultivate a positive atmosphere [109] for customers who engage in browsing shopping items and reviews.

2.2.2 Interpretation of Helpfulness

Review helpfulness maintains a list of aliases, such as review usefulness [180, 98], review utility [339], review quality [172], review informativeness [290], review persuasiveness [238], and review trustworthiness [91, 20]. Oxford Dictionary defines "helpfulness" as (1) the quality of helping in a particular situation and (2) the quality of showing the willingness to help somebody. The Cambridge Dictionary states "helpfulness" is the quality of being helpful. In the context of e-commerce, the helpfulness of a review originates from the collective decision of helpful/unhelpful votes provided by previous users. From a linguistic perspective, helpfulness is an abstract concept [33] and tends to be emotionally valenced [150]. The meaning of helpfulness [68] can be at least understood from two perspectives: broadly speaking, review helpfulness specifies how users evaluate online reviews in practice; in a narrow sense though, helpfulness may only suggest if a review assists one in making a purchase decision. In practice, users vote helpfulness with extremely limited information regarding the definition and/or specifics of the concept. Hence, the interpretation of helpfulness is vague and highly subjective [295] to one's personal knowledge and experiences of and interests in an item.

Several studies exploit human annotation over the voting mechanism to achieve less biased helpfulness interpretation. Still, many provide little [221, 179, 275] or no [127, 337, 322] guidance during the annotation process. The study by Liu et al. [172] is probably the first attempt to alleviate the uncertainty and subjectivity of helpfulness evaluation. Their work provides a clear assessment guideline called SPEC that defines four types ("Best", "Good", "Fair", and "Bad") of review quality based on how a review values users' purchase decisions. The generic description of SPEC is given in the original paper, mainly focusing on the detailedness, relevancy, and convincingness of reviews. The authors employ two annotators to label 4,909 reviews from 100 randomly-selected Amazon digital cameras as per the SPEC instructions. After human annotation, the first three types (i.e., "Best", "Good", and "Fair") are further grouped into high-quality reviews against low-quality (i.e., "Bad") ones.

Chen et al. [47] capitalize on the SPEC guideline to annotate the helpfulness of 19,030 reviews derived from 9,805 Amazon products. Meng et al. [197] follow Liu's paradigm [172] and design a guideline for review text quality measurement. The guideline comprises three core dimensions (i.e., readability, believability, and relevancy) along with several shallow contextual cues such as review metadata and reviewer characteristics. The authors collect 10,235 and 9,607 reviews respectively from top 100 Amazon headset and skincare cleanser products. Using the guideline, nine independent annotators are invited to read each of the reviews and then rate the quality level (either high or low) with at least one supporting dimension. Instead of designating a small group of annotators to mark large-scale of reviews, Tsur and Rappoport [295] employ a large number of annotators wherein each evaluate a few. Compared with [172, 47, 197] using relatively strict guidelines, the authors define a guideline that only loosely describes what makes a good (helpful) book review.

2.2.3 Factors in Human Perception

The last decade has witnessed a number of helpfulness analysis dissecting human perception processes in different domains. These studies offer primary insights into factors that affect human helpfulness assessment.

Hernandez et al. [114] design a questionnaire to factorize human perception in review helpfulness. The questionnaire consists of ten statements describing the characteristics of a review. The authors randomly selected 18 TripAdvisor hotel reviews from the ArguAna dataset [300]. A total of 108 participants hired from Amazon Mechanical Turk are asked to rate the overall review helpfulness using a five-point Likert scale, with 1 (5) being the least (most) helpful. Each participant then needs to inform to what degree he/she agrees the ten predefined statements using another set of five-point Likert scales. Multiple logistic regression on the questionnaire obtains four statements significantly influencing helpfulness voting: a review that (i) addresses hotel aspects that are desired, (ii) is seemingly credible, (iii) includes adequate fact-based objective expressions, and (iv) provides convincing reasons.

Hwang et al. [127] qualitatively identify important helpfulness characteristics from hotel reviews by conducting semi-structured interviews with two senior hotel managers. The authors report three main types of characteristics that the interviewees will extra pay

attention to when perceiving review helpfulness. (i) Topical characteristics that mention experiences on hotel aspects. For example, aspects (e.g., hotel local, hotel decor, travel planning) not directly related to the hotel are less important and usually ignored (interviewee #1), whereas great importance is attached to reviews mentioning hotel services (interviewee #2). (ii) Sentiment characteristics that measure emotional strengths. (iii) Lexical characteristics that reflect writing quality. According to interviewee #1, professional bloggers tend to be prestigious and write meaningful and convincing review content.

Ngo-Ye et al. [221] investigate the contribution of review content to helpfulness via human cognition studies. The authors collect $2,600$ online reviews for 12 items, including $1,381$ from nine Amazon books and $1,219$ covering three Yelp restaurants. A total of 135 undergraduate students at a US business school are recruited to read a set of 8 or 12 reviews describing one particular item. For each review, a participant is asked to: (1) rate the overall helpfulness of the review using a seven-point Likert scale (where one point means not at all helpful and seven considerablely helpful) and (2) annotate words or phrases (also known as concept scripts) in the review that lead to his/her helpfulness rating. The human-perceived scripts prove to be effective in modeling and predicting helpfulness information.

Liu et al. [179] randomly choose $1,000$ online reviews from eight mobile phones on Amazon without any evaluation instructions. Six final-year undergraduates are recruited to rate the helpfulness of all the reviews using a five-degree Likert scale ranging from -2 (least helpful) to $+2$ (most helpful). The authors initiate two questionnaires to investigate the reasons why certain reviews receive unanimous assessment, whereas the helpfulness of some is divergent. The first questionnaire observes that detailed and concrete reviews tend to be more helpful, including (i) long reviews covering customer preferences, (ii) mentions of product features, (iii) mentions of likes and dislikes of a product, and (iv) product comparison. While reasons such as (i) not mentioning pros and cons and (ii) lack of information regarding product performance, are considered less helpful. The second questionnaire further reveals several important perspectives and judgment criteria held by these subjects A review's perceived helpfulness can be increased (decreased) if its described aspects matching (mismatching) the user expectation, namely, information required by a user.

Connors et al. [61] conduct an open-ended field study to understand the essence of helpfulness information. The qualitative perspective askes 40 undergraduate students

majored in business to read 20 online reviews of one single product and then enumerate aspects that lead to their helpful and unhelpful perception. The collected textual aspects are manually refined and grouped based on text similarity, resulting in 18 reasons for helpful reviews and 10 for unhelpful reviews, along with the frequency. Among the reasons, the presence of pros and cons is mostly believed contributing to review helpfulness, whereas the top unhelpful reason is reviews being overly emotional/biased. Most of the unhelpful reasons have their helpful counterparts, including "Good Writing Style" versus "Poor Writing Style", "Lay-Man's Terms" versus "Using Technical Language", "Detail" versus "Not Enough Detail", "Relevancy" versus "Irrelevant Comments". The length of a review is also reported to be highly related to both helpful (e.g., "Lack of Information", "Too Much Detail", and "Too Short") and unhelpful (e.g., "Conciseness" and "Lengthy") reviews. Several reasons belong exclusively to the helpful category such as "Personal Information About Reviewer" and "Comparisons".

As shown, some of the aforementioned factors can be easily adapted into automatic helpfulness prediction. For example, review length can be represented by word count. Adapting the remaining factors, however, is less straightforward and sometimes challenging since many of those concepts (e.g., credibility and authenticity) are subjective and the judgement of such varies between people. Nonetheless, the enlightening findings provide inspiring ideas for future studies.

2.3 Determinants for Helpfulness Modeling

Designing effective features [41, 119] is at the core of helpfulness modeling. Previous literature has largely explored determinants for helpfulness prediction; the determinants can be mainly divided into feature engineering and deep learning methods. The former have long dominated the task by leveraging domain knowledge to manually curate statistics from reviews. The latter adopt neural architectures to automatically learn latent features from reviews, which gains increasing popularity. Section 2.3.1 and Section 2.3.2 respectively describe the two types of feature extraction methods.

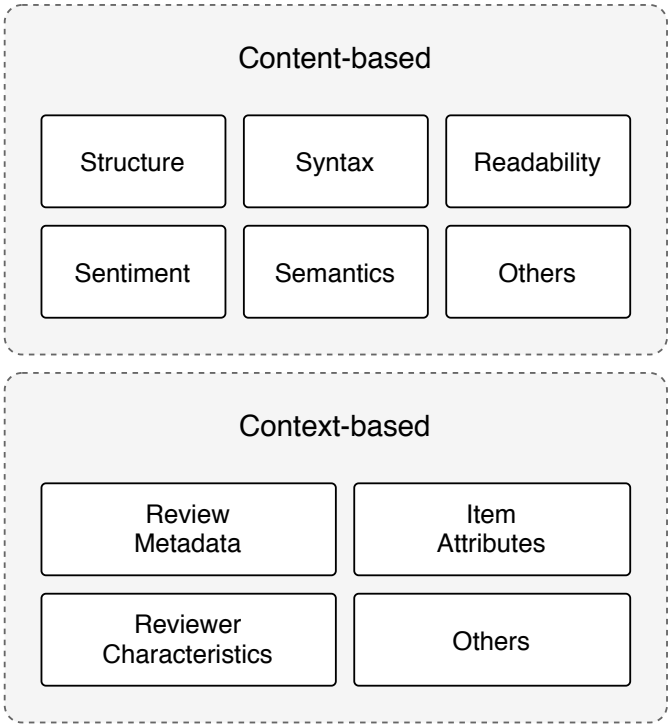

Figure 2.1: Functionality-based hierarchy of hand-crafted determinants.

2.3.1 Hand-crafted Statistics via Feature Engineering

The past decade has witnessed a large body of hand-crafted features [254, 224, 117] carefully curated to represent reviews for helpfulness prediction. The proposed features, although following different naming conventions, can be categorized into content-based and context-based ones. The former include linguistic statistics [141, 165, 192, 322, 152, 189] derived from the textual content of reviews, whereas the latter contain contextual information of reviews [313, 153], reviewers [48, 122], and reviewed items [341, 209]. Both categories can be further sub-categorized by the the proposed features' functionality, as demonstrated in Figure 2.1. While this chapter endeavors to conduct comprehensive survey, it is almost impossible to cover all proposed features. The "Others" subcategories are thus added for the sake of completeness. Note that certain determinants may result from linear transformation of features across subcategories and will be only assigned to one depending on its main focus.

Content-based features

Review texts play an indispensable role in user-generated reviews on almost all platforms. The textual content expresses previous customers' opinions towards a product, which contains arguably the majority information of a review. Content-based features extract various linguistic characteristics from review texts to obtain insights from the opinions. Sorted by implementation complexity, the five coherent subcategories are discussed as follows.

Structure The first subcategory probes into review structure. The structure of a review indicates how reviewers present their comments during review writing. For example, the same comment can be presented as one or multiple paragraphs. Most studies analyze review structure via length (depth) statistics from review texts.

Different granularity levels of language units have been explored to model helpfulness. From coarse- to fine-grained units, the structural information includes the number of paragraphs [21, 172, 221], sentences [21, 299, 191, 155, 177, 110], phrases [242, 172, 141], words [21, 266, 284, 5, 184], characters [21, 299, 122, 160, 177, 110, 221, 97, 161, 28, 229], and syllables [122, 160] in a review. The occurrence of phrases describing product features in a review proves useful, such as "battery life" in smartphones and "image quality" in digital cameras. Kim et al. [141] extract product features from Pros/Cons listings learned from Epinions. The authors in [172] use fuzzy matching [121] to detect product features and brand names, followed by resolution that reduces different equivalence forms of a product feature. Qazi et al. [242] parse and extract concepts (e.g. "wildlife animals", "shuttle bus", and "air condition") from tourism business reviews using the knowledge base SenticNet [37] and compute the average number of concepts per sentence. In [21], Baowaly et al. use (i) the standard deviation of word- and sentence-level counting statistics and (ii) the ratio of review length before to after pre-processing. Structural features can also be exclusively extracted within in review titles [5, 38, 348, 172], subsections of a review/reviewer profile [5], and product description [93]. Given the short nature of online consumer reviews, word- and sentence-level structures are among the most frequently used length statistics.

Many studies combine counting statistics of multiple language units for more informative structural features. For instance, [191, 266, 279, 182] compute the ratio of unique words in a review (also known as lexical diversity or vocabulary richness)

[191, 266, 279, 182]; [5] computes linguistics richness for individual reviews defined as the number of words to that of unique words, including punctuation; [340] computes the ratio of short words (less than four letters) in a review. Another example falls into average counting statistics, such as the average number of sentences per paragraph (average paragraph length) [21, 172, 221, 341], the average number of words per sentence (average sentence length) [21, 47, 299, 226, 191, 155, 127, 161, 38, 221, 171, 321, 179, 341, 172, 322, 182, 141, 122, 160, 73, 343], the average number of characters per word (average word length) [21, 343, 161, 38, 221], the average number of characters per sentence [21, 226], and the average number of syllables per word [122, 160, 73, 343] in a review. [341] calculates the standard deviation of words and sentences among reviews.

Several studies transform continuous length statistics into discrete variables. For each review, [161] and [38] count three types of words: words that have one letter, more than ten letters, and those of length between two and nine letters. [266, 279] summarize the number of one-letter and two-letter words, and those having more than two letters in a review. [84] categorizes reviews into short, medium, and long ones. A review is short (long) if placed one standard deviation below (above) the average sentence length; the rest are considered to be of medium length. Similarly, [320] regards a review as a very short, short, and long one if it has ≤ 48, 49 to 60, and ≤ 61 words, respectively.

Less popular structural features are also used. In [172], the authors design a list of Pros-Cons related concepts (called paragraph separators) and count the occurrence of the concepts in a review. The list contains nouns and noun phrases commonly used by customers to summarize the advantages and disadvantages of a product, such as "The Good", "The Bad", "Thumb up", "Bummer", "Likes", and "Dislikes". Similarly, [331] counts the occurrence of "Pros" and "Cons" in a review. [141] counts two types of HTML formatting tags: bold tags <b> used for emphasis and line breaks
. For each review, [279, 266] computes the number of difficult words defined as those outside a list of $3,000$ familiar words [40]. [73] computes the number of "complex" words (three or more syllables) in a review. The perplexity [226] and information entropy [93, 226, 279, 266] of textual reviews are also considered. In [93], the authors measure the incremental entropy (in terms of word count) of the current review compared with all its predecessors. The first "review" is defined as product description, followed by the second review as the first record in the review list and so on.

Syntax The second subcategory investigates the role of syntax in written reviews. Current studies analyze syntactic components of review texts such as the tenses, parts of speech [175], correctness of spelling and grammar, and sentence patterns.

The distribution of parts-of-speech is largely utilized, particularly the number/ratio of open-class words [47, 343, 318, 177, 182, 141, 192, 189, 191] such as nouns [107, 266, 54, 279, 5], verbs [266, 279], adjectives [5, 266, 179, 279], and adverbs [179, 107]. [339] counts the number of modal verbs and proper nouns, which are usually technical terms, product brands, concepts, etc. [120] designs three degrees (i.e., high, medium, and low) of volitive auxiliaries and calculates the rate of sentences that involve at least one of the auxiliaries. In [141, 318], the percentage of verbs conjugated in the first person is computed. [120] extracts the rate of verbs (in past and perfect tense) linking to product features.

Other parts-of-speech and their combinations include preposition [107], personal pronouns [107], foreign words [191, 182], symbols [191, 182], numbers [191, 339, 182], punctuation [191, 182, 21], interjections [339], modal particles [177], and mimetic words [177]. In [152, 107, 5], the authors extract five manually-coded linguistic categories via the linguistic category model [57], including adjectives, state verbs, and three types (i.e., state, interpretive, and descriptive) of action verbs. [166] measures the ratio of words matching four types (absolute, high, moderate, low) of evidentiality [288]. [275] manually gauges the concreteness (i.e., concrete or abstract) of a review following the definition in [165]. [199] counts explicit discourse connectives in reviews using regular expression to match words and phrases that connect two clauses in reviews, such as “and”, “or”, “but”, “then again”, and “as well as”. [279] counts the number of stop words in a review.

A line of syntactic features measure the extent to which reviews are properly written. Capitalization marks an important indicator, including the number/ratio of capitalized characters and words (commonly used for emphasis) [21, 299, 166, 110], sentences starting with a capital letter [182, 191, 166], uppercase characters [340, 225], and lowercase characters [225]. [225, 47] computes the ratio of uppercase to lowercase characters in a review. [21] checks whether a reviews starts with a capital letter.

The other two factors lie in spelling mistakes [343, 340] and grammatical errors [179]. [166, 162, 161] gather the number/ratio of misspelled words in a review using off-the-shelf English spell checkers. In [97], frequent proper nouns such as brand names

and terminology words are excluded prior to detection. [266, 279] define wrong words as those absent from the Enchant English dictionary. [284] defines review clarity as the percentage of spelling errors captured by Hunspell and grammatical errors computed using the Grammarly API in a review.

Exclamatory, interrogative, and comparative tones can attract more customer attention. [21, 47, 299, 322, 141, 171, 321, 318, 177] count exclamation marks; [177, 322, 141, 171, 321, 318, 47] computes the number/ratio of interrogative sentences; [21, 110, 299] count the number of question marks in a reviews. [339] considers the number of wh-words (i.e., wh-determiners, wh-pronouns, wh-adverbs) that signify interrogation. As for comparative expressions, the number/ratio of (1) comparative adjectives and adverbs [182, 339, 191] and (2) superlative adjectives and adverbs [339] are summarized. [331] matches comparison using two rule-based patterns "compare to/with" and "adjective+er than".

In addition, the argumentative components of online reviews are analyzed. [242] manually labels reviews into regular, comparative, and suggestive ones based on their morphological construct. [171] recruits three independent participants to annotate argument structures of a small sample of reviews. Given a review, each constituent clause is labelled as one of the seven arguments: major claim, claim, premise, premise supporting an implicit claim, background, recommendation, and non-argumentative. Four granularity levels (component-, token-, letter-, and position-level) are constructed upon the annotated argument features. [232] adopts an off-the-shelf argumentation mining system MARGOT [168] to detect sentences and clauses that are claims and premises in review texts. [313] adopts the NET model [65] to extract "subject-predicate-object" triplets from a review, each describing the degree of (dis)association of a subject with an object, for example, "product/positive/ease of use".

Readability The third subcategory measures the extent to which customers read and comprehend online reviews. Even a minor increase in readability largely can improve review readership [79], leading to more opportunities for reviews to receive helpful votes. Seven existing readability (also known as understandability) tests [340] have been frequently used to estimate ease of reading, taking advantage of the structural information of reviews. Although the readability tests are well-researched in English, applying them to other languages may lack statistical validity.

The Flesch–Kincaid readability test conputes the Flesch Reading Ease (FKRE) and Flesch–Kincaid Grade Level (FKGL). Both tests adopt identical core measures (the average number of words per sentence and the average number of syllables per word) but differ in weighting coefficients. FKRE [299, 284, 5, 122, 160, 158, 73, 166, 221, 97, 192, 189, 153, 3, 180, 279, 315, 343, 266, 316] scores are usually between 0 and 100; higher (lower) scores indicate that reviews are less (more) readable. FKGL [5, 122, 160, 73, 221, 97, 148, 152, 189, 343] corresponds scores to United States grade levels, which is extensively used in the field of education.

Four readability tests gauge the years of education needed to understand a piece of writing. The Gunning Fog Index (GFI) [122, 160, 73, 97, 148, 152, 189, 87, 326, 54, 132, 180] confirms that text can be read easily by the intended audience involved in newspaper and textbook publishing. The Simple Measure of Gobbledygook (SMOG) [122, 160, 73, 97, 152, 189, 343] develops a more accurate and easily calculated substitute for FOG. Both tests require counting complex words (of three or more syllables) in a review. Unlike the syllable-based readability indices, the Automated Readability Index (ARI) [93, 299, 5, 122, 160, 209, 97, 148, 152, 189, 230, 54, 180, 276] and Coleman–Liau Index (CLI) [122, 160, 309, 97, 148, 152, 189, 325, 54, 180, 349] aim at faster computation and rely only on readily characters, words, and sentences in reviews.

The Dale–Chall Readability formula (DCR) [279, 266] considers the ratio of difficult words and the average sentence length. The authors prepare a list of 3,000 words (base forms only) that 80 percent of American students can reliably understand. Difficult words refer to those not in the listand if the percentage of difficult words in a review is above five percent, a penalty of 3.6365 is added to the final score.

Table 2.1: Readability tests for helpfulness prediction.

Readability Test	Fomular
FKRE [88]	$206.835 - 1.015\left(\frac{\#\text{Words}}{\#\text{Sentences}}\right) - 84.6\left(\frac{\#\text{Syllables}}{\#\text{Words}}\right) - 15.59$
FKGL [144]	$0.39\left(\frac{\#\text{Words}}{\#\text{Sentences}}\right) + 11.8\left(\frac{\#\text{Syllables}}{\#\text{Words}}\right)$
SMOG [194]	$1.0430\sqrt{\#\text{Complex Words} \times \frac{30}{\#\text{Sentences}}} + 3.1291$
CLI [58]	$0.0588\left(\frac{\#\text{Characters}}{\#\text{Words}} \times 100\right) - 0.296\left(\frac{\#\text{Sentences}}{\#\text{Words}} \times 100\right) - 15.8$
GFI [103]	$0.4\left[\left(\frac{\#\text{Words}}{\#\text{Sentences}}\right) + 100\left(\frac{\#\text{Complex Words}}{\#\text{Words}}\right)\right]$
ARI [281]	$4.71\left(\frac{\#\text{Characters}}{\#\text{Words}}\right) + 0.5\left(\frac{\#\text{Words}}{\#\text{Sentences}}\right) - 21.43$
DCR [40]	$0.1579\left(\frac{\#\text{Difficult Words}}{\#\text{Words}} \times 100\right) + 0.0496\left(\frac{\#\text{Words}}{\#\text{Sentences}}\right)$

The formula of the aforementioned readability tests are as listed in Table 2.1. More details regarding readability computation can be found in the original papers. Different readability tests may return scores of various scale, and interested readers can access the score interpretation in [25, 59].

Sentiment The fourth subcategory employs sentiment analysis techniques to study the valence (i.e., positivity and negativity), subjectivity, and emotion statuses of online reviews. Sentiment features summarize the overall attitude expressed by customers towards a commented target.

Detecting review sentiment can be approached through lexical resources [339]. Common lexicons include the NRC Word-Emotion Association Lexicon (EmoLex) [340, 189, 284, 209, 184], General Inquirer (GI) [153, 322, 171, 321, 318, 141], SentiWordNet (SWN) [158, 127, 15, 279, 152], Opinion Lexicon (OpiLex) [73, 182, 93], Geneva Affect Label Coder (GALC) [322, 171, 192], Linguistic Inquiry and Word Count (LIWC) [21, 322, 321, 72], AFINN [284], WordNet-Affect (WA) [287], and Valence Aware Dictionary and sEntiment Reasoner (VADER) [75]. Depending on usage, the lexicons generate either word valences (AFINN, SWN, OpiLex, and VADER) or emotion categories (GALC and WA); LIWC, GI, and EmoLex provide schemes for both valence- and category-based features. When detecting review valence, negation treatment [127, 179, 192] usually improves quality.

Category-based lexicons can be partly exploited. [318, 276] extract "Positiv" and "Negativ" from GI; [276] chooses "Quality" and "If" from GI to measure the degrees of reviews describing quality-related concepts and uncertainty. Regarding LIWC, [3, 54, 221, 326] extract "PosEmo" and "NegEmo"; [305] defines price cues as the presence or absence of money-related words identified by LIWC; [166] designs four factors (i.e., Immediacy, Making Distinctions, Social Past, and Rationalization) composed of 15 LIWC dimensions; [184, 326] consider words related to reasoning or cognitive mechanism via the "CogProc" dimension; [231] groups 10 LIWC dimensions into two groups: psychological ("Analytic", "Clout", "Authentic, "CogProc", "Percept", "PosEmo", and "NegEmo") and linguistic ("WC", "WPS", and "Compare"). [325, 132] focus on the emotional disclosure interpreted by the "Anxiety" and "Anger" dimensions.

Another approach for review sentiment is to train domain-specific classifiers [160, 97, 341] using machine learning algorithms such as Naïve Bayes [316, 348], Logistic

Regression [184], and Support Vector Machine [343]. The training samples are annotated by domain experts and thus the trained model usually achieve higher accuracy in sentiment detection.

Finally, review sentiment can also be gauged by off-the-shelf tools [84, 5], such as SentiStrength [183, 166, 189, 261] and OpinionFinder [160, 122, 80]. [179] initiates a set of positive and negative adjectives and extends the seed words using WordNet [90] synsets. [166] studies emoticons such as :-) and :-D). [120] computes the difference between a review's valence and the valence shared by majority of reviews. [192] counts the positive and negative word occurrences, using a private-sourced lexicon constructed on hotel reviews.

The detected sentiments have various representations [21, 299, 191, 177, 16]: (1) the overall valence intensity of a review, (2) the number/ratio of positive/negative language units (e.g., words, sentences, latent topics), (3) the number/ratio of objective/subjective (neutral/non-neutral) sentences, (4) the distribution of predefined categories over a review, (5) the sentiment consistency between different measured texts (e.g., review title against review content, product specifications against review content), (6) the one- and two-sideness of review texts, and (7) similar variants and combinations of the above. The threshold that defines positivity and subjectivity may differ depending on applications and domains.

Table 2.2 summarizes the aforementioned sentiment lexicons (predefined valence vocabulary and categories) and tools used for helpfulness evaluation. Interested readers can access further information regarding from the original papers. [253, 329] discuss the applicability of the lexicons and tools and their performance in details.

Semantics The fifth subcategory studies the meaning of review content. The first four sub-categories employ predefined statistics that only roughly describe review texts, which leads to certain information loss. Semantic features analyze customer opinions in a finer-grained manner, by directly modeling different language units (usually words and phrases) in review texts.

BOW models are a standard for review semantics encoding such as unigrams [141, 192, 284, 318, 110], bigrams [141, 192, 110, 284, 318] and trigrams [284]. The rationale behind higher levels of n-grams is that many concepts and product aspects entail compound words; multi-word expressions can improve review representation capa-

Table 2.2: Sentiment lexicons and tools for helpfulness prediction.

Lexicon	Year*	Type	Basic Statistics	Score/Output
EmoLex [205]	2013	Category	141,820 words, 8 categories	—
		Valence	13,901 positive 127,919 negative words	{0, 1} where 0 is negative and 1 positive
GI [129]	1996	Category	11,788 words, 182 categories	—
		Valence	1,915 positive and 2,291 negative words	—
SWN [14]	2010	Valence	155,287 words, 117,659 synsets, 206,941 word-sense pairs	[0, 1] for the percentage of positivity, negativity, and objectivity
VADER [126]	2014	Valence	3,345 positive and 4,172 negative words	$[-4, 4]$ ranging from extremely negative to extremely positive
GALC [268]	2005	Category	267 word stems, 36 categories	—
LIWC [236]	2015	Category	~6,400 words, 93 categories	[0, 1] for the percentage of categories
	2015	Valence	620 positive and 744 negative words	[0, 1] for the percentage of positivity and negativity
WA [287]	2004	Category	4,787 words, 2,874 synsets	Each synset has one or more affective labels[1]
AFINN [223]	2011	Valence	878 positive, 1 neutral, and 1,598 negative words	$\{-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5\}$ ranging from very negative to very positive
OpiLex [121]	2004	Valence	2,006 positive and 4,783 negative words	$\{-1, 1\}$ where -1 is negative and 1 positive
SentiStrength [293]	2012	Tool	—	Multiple outputs[2]
OpinionFinder [255]	2003	Tool	—	$\{-1, 0, 1\}$ for negative, neutral, positive sentiments respectively

* For lexicons/tools with multiple updates, only the latest published version is reported.

[1] The affective labels are "emotion", "mood", "trait", "cognitive state", "physical state", "hedonic signal", "emotion-eliciting situation", "emotional response", "behaviour, attitude", and "sensation".

[2] Four types of outputs are provided: (i) dual sentiment strengths: $\{-1, -2, -3, -4, -5\}$ ranging from not negative to extremely negative and $\{1, 2, 3, 4, 5\}$ ranging from not positive to extremely positive; (ii) a single scale $\{-4, -3, -2, -1, 0, 1, 2, 3, 4\}$ ranging from extremely negative to extremely positive; (iii) $\{-1, 0, 1\}$ for negative, neutral, positive sentiments respectively; and (iv) $\{-1, 1\}$ for negative and positive sentiments.

bility. BOW models mainly encode n-grams with binary values [177, 284], occurrences [284, 232, 73], and the standard TFIDF [141, 192, 284, 232, 318, 343, 340, 110, 299, 21] scheme. [226, 248] calculate the centroid score of a review based on TFIDF. To capture longer range of semantics, [199] constructs dependency bigrams using grammatical dependencies between words.

Several BOW models only use a subset of vocabulary for review representation. For example, [47] skips stopwords and words that occur less than ten times. Similarly, [322, 171, 321] only include words with minimum occurrence of three. Hwang et al. [127] choose approximately $4,000$ words with top TFIDF scores. Zheng et al. [316] instead select the top $3,000$ tokens with highest term frequencies. In [177, 220, 221], the authors employ correlation analysis [106] to select a subset of n-grams. [73] extracts two types of bi-grams following the patterns: (1) a non-sentiment adjective + noun and (2) compound nouns and single nouns that frequently occur with sentiment words nearby. Liu et al. [179] propose a document profile model to extract product features mentioned in a review; the extraction finds frequent co-occurring words measured by Point-wise Mutual Information (PMI) [32]. Zhao et al. [341] manually select nouns and noun phrases that are related to product features and clusters the terms into categories. In [221], script phrases and words highlighted by participants are employed for building semantics. In [165, 275], the authors measure content concreteness [33], namely, the extent to which concrete and abstract words are used in reviews.

In addition, topic modeling learns helpfulness information from semantics. In [38, 47], the authors adopt LSA to discover latent topics from reviews. Luo et al. [183] identify four dining aspects (i.e., Taste/food, Experience, Value, and Location) from online restaurant reviews. Vartika et al. [284] create a lexicon of nine topics (i.e., "Food", "Drinks", "Ambience", "Value", "Service", "Staff", "Cleanliness", "Location", and "Others") from restaurant reviews, covering more than $3,000$ words. Zheng et al. [316] identify five major topics (i.e., Basic Service, Value, Landmarks, Dining, and Core Product) from reviews related to hospitality and tourism. [321] adapts a LDA variant originally used for modeling tweets to extract product aspects. [127] trained two sense-based LDA models on the open-class words in reviews, with the WordNet ontology optionally applied for word sense disambiguation. When training topic models, the number of topics [316, 343, 21, 213] can be set based on domain-specific experience or learned automatically.

Shallow neural networks such as Skip-gram model [201] encode helpfulness-related

semantics into distributed representations. Baowaly et al. [21] train 1000-dimensional word vectors from $2,251$ reviews collected from the online gaming platform Steam. In [47], the authors learn 100-dimensional word embeddings on 5.8 million Amazon product reviews. Haque et al. [110] vary the vector length from 5 to 100 with increment by 5. [75] computes the representation of a review by averaging the embedding of its constituent words; review vectors can also be computed by learning along with word embeddings. In [184], Bernhard et al. employ the Paragraph Vector model [157, 66] to directly learn embeddings for each sentence of a review, which are latter used for inferring the two-sideness of review sentences.

Text similarity estimates the closeness between reviews in semantic meaning. Cosine similarity is the most used option, which computes the cosine of the angle between two review representations. Zhang et al. [339] respectively measure the similarity between the TFIDF representation of a customer review and (i) that of the product specification and (ii) that of the editorial review. Zhou et al. [348] vectorize reviews via TFIDF and calculate the similarity between a review's title and its content. In [266], the similarity between review texts and product description [343], and that between review texts and questions discussed on the product page are calculated. Zhang et al. [340] compute the similarity between a current review and its preceding one.

Kullback–Leibler (KL) divergence measures similarity by interpreting the difference between two probability distributions. [182] obtains the divergence between each review and an "average" review that contains all review texts of an item, using unigram language model for text representation. Similarly, Maroun et al. [191] obtain the divergence between the vector representation of one review and all reviews of a product. The rationale is to measure to what extent a review conforms to the general opinion. Several studies employ multiple methods to measure text similarity. Jahna et al. [226] calculate text similarity between a review and product description using (i) cosine similarity, (ii) bigram overlap rate, and (iii) the length of the longest common sub-sequence normalized by review length. In [331], Zeng et al. calculate the number of unigrams, bigrams, and trigrams shared between a review and product description. The authors further define a concept called "degree of detail" to infuse the three types of n-gram features. Hong et al. [120] capture the overlapping rate of product attributes and functions between a review and the editorial review of a product.

One can refer to the original papers and related survey papers [6, 67] for more details regarding different weighting schemes [190], topic modeling techniques [29], embed-

ding learning models [36], and their variants. Section 2.3.2 will discuss learning deeper review semantics via more advanced neural architectures.

Context-based features

Besides review texts, review context is a vital ingredient in successful helpfulness composition. The contextual information reveals cues such as review metadata, reviewers' characteristics and idiosyncrasies, and inherent attributes of shopping items. Review context validity and content-context consistency can also affect the perceived helpfulness. The three coherent context-based subcategories are discussed as follows.

Review Metadata The first subcategory describes metadata of a review or an item, including quantitative evaluation and temporal/spatial logs. Such data help complement the understanding and validation of a reviewer's opinion, and thus helpfulness.

Prior literature has largely employed review star ratings [349, 320, 107, 284] as a quantitative complement to qualitative text description. Current rating mechanism often leverages five-point Likert scales (ranging from "strongly dissatisfied" to "strongly satisfied") to quantify reviewers' overall attitudes toward items and/or that toward item aspects. For example, O'Mahony [225] consider a series of review sub-ratings (e.g., "Rooms", "Cleanliness", and "Business Service") of Las Vegas and Chicago hotels on TripAdvisor.

Linear star ratings (i.e., raw rating values) [348, 93, 305, 72, 299] mark the primary form of rating information. In [49, 341], the authors obtain the fraction of one- and five-star reviews in an item. Ghose et al. [97] check whether a review receives a moderate (three-star) rating; the same concept is called "equivocality" in [137]. Park et al. [230] divides ratings into positive (four- and five-star) and negative (one- and two-star) ones. Chua et al. [54] split reviews into three groups: favorable and unfavorable ones if five- and one-star ratings are given, and mixed ones otherwise. Average star ratings are also popular in helpfulness modeling, covering a wide range of products [161, 229, 141, 184, 162, 341, 97], hotels [158, 349, 189, 314, 225], restaurants [132], mobile applications [137, 326], merchants [325], and peer-review papers [318]. Chevalier et al. [49] capture the annual change in average star ratings, the cumulative number [325, 326], and standard deviation [341, 326, 225] of review ratings for an item. Hu et

al. [122] record the number of reviews having the same rating as a newly-added review. Raw rating ratings can be normalized [73, 77] into values between 0 and 1.

Rating extremity [299] has also been largely studied. Extremely positive reviews may be due to product promotion, whilst extremely negative ones damning reviews from business competitors. One's opinion deviating from the general opinion [16] can influence helpfulness perception. As such, [184, 213, 153, 152, 343, 349, 221] subtract a review's rating by the average rating of an item to measure how and how much the former deviates from the latter; [326, 38, 314, 141, 209, 162, 318, 276] focus on only the magnitude of the divergence is considered. Rating inconsistency can also be measured via other agents. Dong et al. [73] extract topics from review texts and calculate the difference and absolute difference between a review's rating and the proportion of unique positive topics. Bernhard et al. [184] compute the dispersion of an item's average rating and its price. Several studies capture the U-shaped relationship [212] between review ratings and helpfulness, using the square of [284, 180, 325, 230, 348] of linear star ratings. In [309], Wang et al. only consider the quadratic term of moderate (three-star) rating and two most extreme (one- and five-star) ratings.

Review age [107, 343, 5, 305], usually in the form of days [161, 349, 73, 122, 38, 229, 179, 226, 87, 313, 326, 221, 314, 21, 348], weeks [132], and years [184], reflects how long a review has lasted since published. Review age gauges the passed time of a review up until a certain timestamp (usually the date of data collection). Siering et al. [276] define review age as days that have passed since January 1, 1960. Krishnamoorthy et al. [152] compute the number of elapsed days since the release of a product. Hu et al. [122] count the number of days in which a review stays on the first page of the review list. [349, 213] compute the time difference between the publication date of the first review and that of the current one to cope with the early bird bias [172]. To reduce age skewness, Salehan et al. [261] apply a logarithmic scale to the number of elapsed days. In the movie domain, Liu et al. [175] learn the relationship between review age and helpfulness using an exponential decay model. In the work, review age compares the date a review is written with that a movie is premiered. Instead of gauging the number of days, Malik et al. [189] probe into the week day a review is submitted.

Other types of review metadata are as follows. [48, 284] count the number of user-uploaded images along with a review. Yang et al. [320] design three sub-dimensions ("None", "Hotel-related", and "Unattractive hotel-related") for photos added in online hotel reviews. Liu et al. [177] detect whether a review contains pictures and the times a

review is viewed and replied. The authors also estimate the perceived enjoyment of Yelp restaurants in London New York City through the received number of funny and cool votes of a review. Kwok et al. [155] inspect the presence of manager responses to $1,405$ Texas hotel reviews on TripAdvisor. Baowaly et al. [21] examine whether a review (i) contains external links, (ii) mentions game scores, and (iii) belongs to the top entries in the review list. [165] simulates an online shopping environment to study product review helpfulness. The authors generates four types of reviews based on two authorship sources (e.g., expert-written and customer-written) and two content abstractness levels (e.g., abstract and concrete). Wu et al. [314] check if a review of an Amazon product is (i) the most helpful/critical one, (ii) within the top 30 rankings, or otherwise. The ranking threshold is set according to [280], which states reviews within the range largely stabilize over time. In [80, 314], the authors mark whether a review is connected to a verified purchase and validate the completeness of reviews. Cao et al. study reviews on the online software market CNET and check whether the (i) "Pros", (ii) "Cons", and (iii) "Summary" section in a review is filled. O'Mahony et al. [225] check the number of optional blanks being filled when writing a TripAdvisor review: (i) liked and disliked parts, (ii) personal and purpose of visit details, and (iii) template questions.

Reviewer Characteristics The second subcategory analyzes reviewers' demographics, registered information, and historical activities/behaviors. Such data allow future readers to detect whether (i) a reviewer is normal or suspicious, (ii) the author is experienced in the domain, and (iii) reviewers share similarity.

Reviewer credibility [97] relies on user-provided personal information in their profiles such as the name (e.g., user name, real name) [226, 16, 3, 230, 313, 284], avatar [137, 284], age/date of birth [155, 160, 123], gender [137, 155, 160, 284], and location [313, 180, 5] of reviewers. Yang et al. [320] specify four location options (i.e., "Not given", "Foreign", "Domestic", and "Local") and four levels of accumulated helpful votes (i.e., "Least voted", "Less voted", "More voted", and "Most voted"). Lee et al. [160] detects whether a reviewer is a foreigner. [155, 160, 123], categorize a reviewer's age into seven intervals following the settings in TripAdvisor: 12 years and under, 13-17 years, 18-24 years, 25-34 years, 35-49 years, 50-64 years, and 65 years and over. Bjering et al. [28] inspect whether a reviewer is associated with a verified purchase. Liu et al. [180] check whether reviewers use (i) full names or initials and (ii) clear avatars revealing their faces. Karimi et al. [137] examine the legitimacy of reviewer names on

Google Play. The disclosure or concealment [5, 28, 276, 162, 284, 80] of the personal information also affects reviewer credibility.

Reviewer expertise [230, 48] studies reviewer contribution levels [177] and rankings [158, 97, 161, 162, 226]. A line of work [155, 160, 122, 320] models TripAdvisor contributor levels into helpfulness, which are linked to the number of posted reviews. As for the Yelp platform, Li et al. [132] check whether a reviewer is titled as an Elite; [349, 230, 180] count the number of Elite badges (awards) a reviewer owns. Many studies [16, 340, 226, 97, 313, 5] validate whether a reviewer is a top-ranked Amazon user. Siering et al. [276] measure reviewer experience through the logarithm of Amazon rankings. Subhabrata et al. [213] capture reviewer expertise on Amazon evolving over time via a Hidden Markov Model. Akbarabadi et al. [5] check (i) whether a reviewer has received any Amazon badges and (ii) whether a reviewer is a member of the Amazon Vine Program.

The reciprocity within online peer-reviewed platforms often measures (i) the total number of reviews a reviewer has contributed to a platform [97, 124, 225, 182, 189, 226, 87, 3, 230, 180, 127, 162, 177, 284, 305, 5, 160, 191], and (ii) that of votes [97, 124] and helpful votes [97, 189, 226, 305, 5, 160, 158] a reviewer has received from the platform. Common variants include (i) the average [213, 97, 127, 225, 160, 284, 213] and standard deviation [127, 225] of the number of helpful votes and (ii) the percentage of helpful votes [161, 124, 162]. Liu et al. [177] count the number of review replies posted and received by a reviewer. O'Mahony et al. [225] further compute the ratio of a user's reviews that have at least five votes and the mean and standard deviation of the number of individual users' reviews.

Reviewers' rating behaviours is also studied. For example, the mean [87, 160, 155, 191, 225, 182], standard deviation [225], and skewness [87] of a reviewer's historical star ratings are computed to measure his/her rating tendency and consistency. Local rating deviation [87, 161, 349] gauges the extent to which a reviewer' current rating is similar to his/her usual rating behavior. Global rating deviation [213, 213] gauges the extent to which a reviewer's rating behavior differs from the general community. The former (latter) computes the absolute deviation between a reviewer's rating on an item and the reviewer's mean rating over all other items (the mean rating of all his/her available reviews on the platform).

Several online platforms allow for rich social functions and the social context [185,

204] can be integrated into review helpfulness for regularization. [291] explores the trust network in an online community to model the connection between individual reviewers and raters. Given an entity (i.e., a reviewer, rater, or both) in the network, the number of its trustors (also known as followers, fans, and friends) [191, 284, 349, 132, 48, 180] and trustees [191] are modeled. [191, 182] compute the PageRank score [227] of each entity. Lu et al. measure the the in/out-degree of the entities. Maroun et al. [191] calculate the average rating given by the entity's similar users, trustors, and trustees. To model the relationship between a reviewer and rater, the authors further explore (i) products they rated in common (ii) their common trustors (trustees), (iii) paths from one to another in the trust network, and (iv) the similarity between their historical ratings,

Other rarely used features are as follows. Malik et al. [189] defines activity length as the number of days between the first and last review authored by a reviewer. Activity length also measures the lapsed days [122, 160], weeks [132], and years [155] since a reviewer has joined an online platform. For example, Vartika et al. [284] obtain the number of years since a reviewer has registered a Yelp account and has become a Yelp Elite member. Spatial statistics are also used, such as the number of cities a reviewer has visited [155, 160] and his/her total travel distance (in miles) [160]. [160, 220] propose a three-dimensional feature that contains (i) Recency: the date difference between the last and the second-last review authored by a reviewer, (ii) Frequency: the total number of reviews of an item, and (iii) Monetary: the total number of reviews the reviewer has posted and votes the reviewer has received. Baowaly et al. [21] record the total number of games a review owns and hours he/she has spent on a game prior to writing the review. Pan et al. [229] conduct content analysis and construct 21 binary attributes closely associated with the innovativeness [256] of a reviewer.

Item Attributes The third subcategory focuses on inherent properties of an item, which are more user needs (e.g., brand reputation) related, domain-specific, and platform-specific. Although not directly composing any parts of online reviews, the attributes still influence users in perceiving review helpfulness.

Item popularity is frequently discussed. Many studies draw from the idea that popular items can attract more visits/purchases and thus feedback. The cumulative number of reviews/photos per item has been widely used for products [161, 309, 162, 341], hotels [225, 97, 229, 349, 158, 225], books [49], mobile games [137], attractions [87], to name a few. Chevalier et al. [49] consider both (i) the fraction of books with no reviews

and (ii) the annual review increment of a book. [225] computes the mean and standard deviation of the number of reviews across hotels. Ranking and temporal statistics can also measure popularity. Fang et al. [87] consider the ranking of New Orleans attractions collected from TripAdvisor. Yin et al. [326] collect and rank applications from the App Store based on the number of downloads. In [97] and [161], the authors record the number of days since the release of a product. Similarly, Zhou et al. [348] compute the elapsed time from the date of the first review posted for a product.

The economic impact of items is another factor, for example, sales [9] and prices [326, 49]. [49] extracts the sales rank of books both available on two online bookseller websites: Amazon and Barnes & Noble. Zhao et al. [341] obtain both product sales and retail price statistics spanning more than 18 months. Zhu et al. [349] collect the price range (from $ to $$$$) of all San Francisco hotels from Yelp. Li et al. [132] count the number of reviews of each price level from all Yelp restaurants in the Phoenix city. The sales and price statistics [313, 189, 97, 161] are also collected from different Amazon product categories.

The nature of an item is yet another factor. Many studies [80, 28, 16, 276, 162, 309, 313, 3] divide items into experience and search products [212], based on (i) the ease of product information acquisition and (ii) the dependence of one's senses for objective product comparison. In [229], Pan et al. refer to the two types as experiential and utilitarian products, whereas Mousavizadeh et al. [209] opt for expressive and functional ones, respectively. [16] further distinguishes high- versus low-priced goods.

Other relatively rare factors are as follows. Erin et al. [305] classify hotels into high-class (5- and 4-star) and low-class (3- and 2-star) ones based on the hotel star levels [122] defined in TripAdvisor. Zheng et al. [316] collect reviews of Manhattan hotels from TripAdvisor, Expedia, and Yelp, and create a categorical variable to indicate the information provider (i.e., source) of a review. In [175], a movie is represented by a binary vector wherein each dimension indicates whether the movie belongs to a certain genre. Chevalier et al. [49] record product shipping time options, including "Usually ships in 24 hours", "Usually ships in 2–3 days", and "Usually ships in 1–2 weeks". [21] inspects (i) whether a game would be recommended by a reviewer to other users and (ii) the genre of the reviewed game. Yin et al. [326] study (i) whether an application releases an update, (ii) whether the application is compatible with iPad, (iii) the application size in megabyte, and (iv) the number of applications developed by the maker.

2.3.2 Neural Features via Deep Learning

The emergence of deep learning techniques has brought success into many research areas and applications. Compared with heavy feature engineering, neural architectures can automatically learn latent features from both review content and context, while avoiding error propagation. Recent studies start to extract helpfulness-related features in an end-to-end manner. Hence, deep learning can better capture factors and their moderating effects that lead to helpfulness perception in reality.

Several primitive studies apply neural networks in place of traditional machine learning algorithms. Multilayer Perceptrons (MLPs) [269] with one hidden layer are largely used due to its simplicity. [84] trains two hidden neurons (determined by the validation set) to predict helpfulness. [161] trains 20 hidden neurons, each corresponding to one of the 20 predefined determinants encompassing helpfulness. [22] trains five nodes in the hidden layer on 11 hand-crafted features to analyze the helpfulness of reviews written in Brazilian Portuguese. [72] selects review ratings and a subset of LIWC dimensions as features across five categories, which are fed into MLPs with two hidden layers. Still, the manually-prepared features require tremendous efforts. This chapter defines deep learning methods as those learn features in an end-to-end manner. In other words, only inputs and outputs of a model is specified and the learning of latent feature is (almost) free from human intervention.

More recent studies focus on end-to-end helpfulness prediction. Text content contains arguably the richest information of a review, which serves as an ideal network input. Depending on applications, additional data inputs and outputs such as social relationships and review ratings are also considered, together modeling the perceived helpfulness of reviews.

[267] adopts a two-layer CNN framework. The first convolution layer converts word embeddings of a review into a document-level representation. The second convolution layer further encodes the review for helpfulness prediction. The authors tune the model over a series of hyperparameters, including filter regions (tri-gram, four-gram, five-gram, and the all combined), dropout rates, epoch size, and batch size. Experimental results show that the presented model learn more complex and accurate semantics than hand-crafted features.

[44] considers a cross-domain task that seeks to predict review helpfulness of one

domain with limited training data, using knowledge learned from another with sufficient reviews. The CNN framework first enriches word embeddings with subword information, aiming to alleviate the out-of-vocabulary problem in many applications, especially when training data is insufficient. The authors build three separate CNN layers on the document embeddings to perform knowledge transferring: one summarizes common knowledge shared across domains and the remaining two learn domain-specific knowledge. To ensure each CNN layer learns the corresponding knowledge, adversarial loss is defined on the domain-independent layer, whereas domain discrimination losses and negative cross-entropy losses are added to the rest.

[43] extends [44] to study multi-domain helpfulness prediction. Inspired by observations that helpful reviews tend to mention certain product features, the authors incorporate aspect distribution of reviews [321] into word embeddings to learn review representations. The enhanced word embeddings are then fed into a domain specific gating layer (a feed forward layer with element-wise sigmoid activation) to identify (un)important words in reviews before convolution. Similar to [44], a shared and domain-specific layer is added to model domain commonalities and differences, respectively. During training, the heterogeneous relationship among domains is captured via a domain correlation matrix.

[245] proposes two CNN variants to integrate star rating information into helpfulness prediction. The first variant follows traditional machine learning methods and regards a star rating as an extra dimension and concatenate the raw rating and encoded document embedding. The second variant embeds star ratings to have the same dimensionality as word embeddings. Inspired by humans evaluating review texts and star ratings at the same time, the two embeddings are concatenated for document embedding learning.

Instead of combining star ratings into the learned features, [85] trains a multi-task learning neural network for helpfulness and star rating prediction. Similar to [44], review content representation is enhanced by subword information. After convolution, each kernel transforms the word embedding matrix into a feature map. The authors then compute the weighted average of the vectors via attention mechanism. The learned review representation is used for two prediction tasks: the prediction of review helpfulness treated as a classification problem, and that of the accompanying review star rating treated as a regression problem.

[86] considers that review helpfulness should be evaluated within the context of

product characteristics. The assumption lies in that a review tends to be more helpful if it contains information mentioned in the targeted product title. The authors propose to learn product-aware review representations. First, two sets of bidirectional LSTMs are used to learn separate representations for review content and the product title. To establish awareness, the authors match the product title against reviews on a word level via attention mechanism to reinforce review representations. The attention weights reflect the closeness between review content and product characteristics. The product-aware review representations are fed into the penultimate layer for helpfulness prediction following the training objectives in [85].

[187] investigates the role of visual cues in the hotel industry, particularly, the extent to which photos posted along with reviews influence review helpfulness. Using Yelp and TripAdvisor as examples, the authors propose a neural model to encode respectively review texts and the accompanying photos. Text representations refer to the last time step of a vanilla LSTM trained on the text content. Image representations are obtained via a pre-trained 152-layer deep residual network [112]. Both learned representations are then concatenated and fed into another vanilla LSTM network to mimic the reading process. The performance across platforms concludes that user-provided images alone offer little insight into helpfulness evaluation, but consistently strengthen and reinforce the written content.

[233] employs a deep dynamic CNN [135] to estimate review helpfulness and recommend representative reviews in a real-world scenario. Specifically, is employed to estimate review helpfulness. Different from the original CNN framework, the authors repeat twice wide convolution operations followed by dynamic pooling on word embeddings to obtain a higher abstract level of feature map. Top results predicted by the model are recommended. To maximize recommendation diversity, the authors extract aspect-sentiment tuples from review texts and categorize them into a predefined product catalogue hierarchy based on semantic similarity. A review is selected only if its product aspects and the corresponding opinions increase the coverage of those of the entire review set.

[45] couples helpfulness evaluation with the prediction of review star ratings. The prediction framework integrates features learned from review texts into a latent factor model; the content-based features are learned similarly to [342], which consists of two parallel neural networks for modeling user- and item-related reviews. The prediction layer then combines and interacts the two networks; the framework aggregates the en-

coded vectors by their importance learned using a two-layer attention network. After training, the learned attention weights are used as a guideline for measuring, ranking, and recommending useful reviews.

Ge et al. [96] extend [45] from two perspectives. Inside the user/item encoding modules, hierarchical attention networks are employed to select important words and sentences from relevant reviews to learn user/item representations. The inferred attention weights in [45] are a proxy of helpfulness measurement, which might not reflect the reality perceived by customers. To better utilize the collective wisdom, the authors train a helpfulness discriminator on a subset of reviews with voting data and then integrate the discriminator into the prediction framework for attention-based rating regression. The helpfulness discriminator outputs are further transformed into a query vector for the attention network, which can be thought of as refinement extracting helpful elements in the representations.

2.4 Helpfulness Labelling

Predicting helpfulness on new reviews requires knowledge learned from previous ones. To this end, predictive models are first trained on labelled reviews to extract and map representative features of each review to their estimated helpfulness and then used to fulfill the task.

Helpfulness labelling often depends on a review's received votes [7] in the form of "X of Y people think a review is helpful".

The "X of Y" and "X" helpfulness [7] are two widely used estimation methods. The former computes the percentage of positive votes received by a review, whereas the latter leverages the raw "Yes" votes. Zheng et al. [316] transform reviews into TFIDF representations and compute the average opinion as the centroid of the vectors. The helpfulness score is then measured by the cosine similarity between each review and the centroid. Similarly, Jorge et al. [93] design helpfulness by computing the cosine similarity between the LSA representation of a review and that of the term "helpful".

Continuous helpfulness values can be converted into categories to make the estimation more intuitive and straightforward for customers. The standard scheme is dichotomous discretization on the "X of Y" helpfulness. Given a threshold, all reviews

are marked as either helpful or unhelpful. Ghose et al. [97] found a threshold equal to 0.6 balances the false positive and the false negative rate between human annotation and voting data. A large body of studies have adapted the threshold [152, 189, 343, 5]; studies [245, 85, 86, 171, 73, 162, 232, 21] also manually select the threshold within the range $[0.5, 0.9]$. Several studies trichotomize continuous helpfulness. Mertz et al. [199] initially consider reviews having the top and bottom 30% helpfulness respectively as bad and good ones and then adjust thresholds to ensure both classes have approximately equal size. The middle part of the reviews are treated as uncertain and eliminated to improve voting data reliability. Similarly, Sheng-Tun et al. [166] consider reviewers whose average review helpfulness values fall in the top (bottom) 40% as helpful (unhelpful); the remaining reviewers are discarded to minimize possible voting biases. Martin et al. [192] pursuit higher review quality by only selecting helpful reviews as those yielding the top 1% helpfulness. To cope with reviews with high confidence but little support, Zeng et al. [331] develop a log-support scoring method based on the voting data, on which three classes are defined: (i) the helpful positive reviews, (ii) the helpful negative reviews, and (iii) the unhelpful reviews. For voting data where only "Yes" votes are available, small numbers [187, 50] are often chosen as thresholds. Saumya st al. [266] set the threshold as the average helpfulness over reviews of each product.

The current helpfulness labelling methods raise several concerns. First, review voting is susceptible to factors such as website layouts and other biases [38, 68]. Second, many reviews do not receive statistically adequate votes [322, 316] to perform reliable helpfulness estimation [291]. Third, a review's votes may inaccurately reflect user feedback [47] since not every customer that reads reviews will vote [141, 22]; the number of people without voting actions is unavailable [22] for collection. As a result, the frequently employed "X of Y" measurements are likely to overestimate [291, 22] helpfulness when reviews have extremely few votes, which is known as "words of few mouths". To handle the issue, [334, 340] estimate review helpfulness through Bayesian inference from the original voting data. Recently, Jardeson et al. [22] induce a reviewer's votes, using the lower bound of Wilson score confidence interval for a Binomial parameter [4]. Some websites such as Reddit[1] and Yelp[2] have already used the same technique to order their posts and reviews.

Finally, review helpfulness can be estimated via manual annotation [322]; the an-

[1] https://redditblog.com/2009/10/15/reddits-new-comment-sorting-system/
[2] https://blog.yelp.com/2011/02/the-most-romantic-city-on-yelp-is

notated helpfulness, again, can be numeric and/or categorical. Human annotation is ideal for obtaining high-quality and less biased helpfulness estimates immune from social influences. Because the annotation task can be laborsome, time-consuming, and expensive, the annotated reviews are often restricted to small data size.

2.5 Helpfulness Learning Sources

Online reviews used for helpfulness prediction are either collected from primary or secondary sources. The former refers to (i) writing computer programs called crawlers or (ii) using application programming interfaces that directly scrape data from targeted platforms. Such methods allow for high customizability and access to up-to-date review information, but can be challenging in terms of time and expense; thus the size of the collected data is usually small. The latter refers to off-the-shelf datasets prepared by previous researchers. These datasets usually possess public accessibility and larger data size for exploring various models; the pre-collected reviews, however, may suffer from timeliness and not reflect the latest trend of several item types.

Most existing studies focus on primary data [224], also known as ad-hoc datasets. Currently, reviews from three online platforms—Amazon [212, 148, 3], Yelp [230, 349, 183], and TripAdvisor [158, 160, 122]—are more favoured among the research community due to their influence in reality. These reviews mainly cover user-generated opinions towards a variety of products, hotels, restaurants, and attractions. Other review sources include software programs [38] on CNET, electronics [325] collected from Yahoo! Shopping, mobile applications on App Store [326] and Google Play [137], movie reviews [175] on IMDB, books [49] on Barnes & Noble, automobiles [177] from AutoHome, video games [22] on Steam, to name a few. Some reviews are obtained from private sources. For example, Zhao et al. [341] construct a longitudinal dataset from an e-commerce company selling girls' clothes. Ad-hoc datasets are seldom shared with the public, with few exceptions [93, 347, 225]. The data unavailability is one of the major reasons that largely obstructs result reproduction and comparison.

The past few years have seen a raising trend of open-source repositories [21, 321, 322, 347] for helpfulness analysis. Chen et al. [47] leverage Amazon review data proposed by [131], which was originally designed for opinion spam detection. Two more popular alternatives include the Amazon Multi-Domain Sentiment Dataset

[120, 152, 189, 107, 245] and the Amazon Review Data [184, 5, 232, 299]. The latter is more recently released, containing extra (and newer) reviews, product categories, and detailed metadata. The annual Yelp Dataset Challenge [192, 132, 284] maintains a large scale of review dataset for public access. To the best of the author's knowledge, only few studies [171] employ pre-collected TripAdvisor reviews for helpfulness learning, probably because hotel industry requires the latest reviews to gains timely insights. Some sources such as Ciao [191, 291, 182] and Epinions [204, 166] (both platforms are now defunct) feature social networking and enable trust connections. Table 2.3 briefly summarizes the descriptive statistics of the aforementioned sources.

Table 2.3: Public datasets for helpfulness prediction.

Dataset	Year	#Reviews	#Items	#Categories	Voting*
Amazon Multi-Domain Sentiment Dataset [31]	2009	~1.4 millon	246,505	25	Binary[a] (X of Y)
Amazon Review Data [195]	2013	~34.7 millon	~2.4 millon	28	Binary (X of Y)
Amazon Review Data [113]	2014	~142.8 millon	~9.4 million	24	Binary (X of Y)
Amazon Review Data [222]	2018	~233.1 million	~15.5 million	29	Binary (X)
Yelp [324]	2019	~6.7 million	192,609	1300	Binary (X)
TripAdvisor [225]	2010	225,936	51,635	2	Binary (X of Y)
Ciao [291]	2013	304,545	112,838	69	Six-point scale[b]
Epinions [292]	2011	~1.3 million	341,596	36	Five-point scale[c]
Epinions [193]	2007	~1.6 million	200,953	—	Five-point scale

* Voting behaviors among the platforms may change over time. Some platforms display voting results in the form of "X of Y", whereas others only present the yes votes X.

[a] 0–Not Helpful; 1–Helpful.

[b] 0–Off Topic; 1–Not Helpful; 2–Somewhat Helpful; 3–Helpful; 4–Very Helpful; 5–Exceptional.

[c] 1–Not helpful; 2–Somewhat Helpful; 3–Helpful; 4–Very Helpful; 5–Most Helpful.

Several pre-collected datasets [164, 282, 306] designed for other research purposes are potentially suitable for helpfulness learning. In addition, there remains a series of review collections ready for use on data science related online communities such as Kaggle[3]. Given the diversity of e-commerce, many interesting portals and item types are still open for exploratory research.

[3] `https://www.kaggle.com`

CHAPTER 3

FEATURE IDENTIFICATION AND SELECTION FOR HELPFULNESS PREDICTION: AN EMPIRICAL STUDY

Online product reviews underpin nearly all e-shopping activities. The high volume of data, as well as various online review quality, urges for automatic approaches for informative content prioritization. Despite a substantial body of literature on review helpfulness prediction, the rationale behind specific feature selection is largely under-studied. Also, the current studies tend to concentrate on domain- and/or platform-dependent feature curation, lacking wider generalization. Moreover, the issue of result comparability and reproducibility occurs due to frequent data and source code unavailability. This chapter addresses the gaps through the most comprehensive feature identification, evaluation, and selection. To this end, the 30 most frequently used content-based features are first identified from 149 relevant research papers and grouped into five coherent categories. The features are then selected to perform helpfulness prediction on six domains of the largest publicly available Amazon 5-core dataset. Three scenarios for feature selection are considered: (i) individual features, (ii) features within each category, and (iii) all features. Empirical results demonstrate that semantics plays a dominant role in predicting informative reviews, followed by sentiment, and other features. Finally, feature combination patterns and selection guidelines across domains are summarized to enhance customer experience in today's prevalent e-commerce environment.

3.1 Introduction

Customer product reviews play a significant role in today's e-commerce world, greatly assisting in online shopping activities. According to a survey conducted in 2016 [214], 91% of online shoppers read product reviews while searching for goods and services, and 84% of them believe that the reviews are equally trustworthy [39] as recommendations from their friends. Online reviews do not only enhance the customer purchasing experience through valuable feedback provision, but also facilitate future product development activities by better understanding the customer needs.

Online product reviews are also highly susceptible to quality control [207], which can potentially harm online shopping experience. A recent study [13] shows that users tend to limit their attention to only first few reviews, regardless of their helpfulness.

It is generally viewed that helpful reviews have more impact on customers' final decisions. However, the large and overwhelming nature of online product reviews makes it difficult for customers to efficiently locate useful information. Although the majority of online platforms enable review helpfulness assessment through user voting, the large proportion of records does not contain any votes. The scarcity of user votes is even more noticeable for less popular products.

Automatic helpfulness prediction helps consumers identify high-quality reviews, which has attracted substantial attention. The mainstream approach follows a procedure of careful feature curation from multiple data sources [224]. Still, the features are frequently domain- and/or platform-dependent, substantially inhibiting wider application. Also, the features are selected arbitrarily without solid justification. Furthermore, prior research mainly focuses on the predictive power of the entire feature set, while little is known on the contribution and necessity of using individual or subsets of features. Since identical feature set is rarely used among existing studies, the reported results prove challenging for fair comparison. Finally, the existing studies are often conducted on publicly unavailable ad-hoc datasets, hampering result reproducibility.

To address the aforementioned gaps, this chapter comprehensively identifies, evaluates, and selects representative features for helpfulness prediction. Specifically, frequently used domain- and platform-*independent* features (i.e., content-based features) are first identified from considerable recent literature. The predictive power of the identified features is then evaluated on six domains of large-scale online product reviews. Instead of evaluating the entire feature set, this chapter allows for performance-oriented feature selection under multiple scenarios. Such flexibility can effectively justify (not) selecting certain features. As a result, feature combination patterns and selection guidelines across domains are summarized, offering valuable insights into general feature selection for helpfulness prediction. The publicly available source code and datasets ensure result comparability and reproducibility of the chapter.

This chapter contributes to existing literature in four aspects:

- First, the chapter conducts one of the most comprehensive literature reviews on helpfulness analysis to identify frequently used content-based features.
- Second, the chapter conducts the first and most extensive empirical validation on large-scale publicly available online product reviews to report feature behaviors in multiple scenarios (individual and combinations) and domains.

- Third, a holistic computational framework is developed for helpfulness prediction from scratch, including data pre-processing, extracting the identified features, and evaluating the predictive power of individual features and feature combinations.
- Fourth, the source code, dataset splits, pre-processed reviews, and extracted features have been released for result reproducibility, benchmark studies, and further improvement.

The remaining of the chapter is organized as follows. Section 3.2 systematically surveys recent literature regarding current features and feature selection strategies for automatic review helpfulness prediction. Section 3.3 introduces the computational framework for feature-based helpfulness prediction, including steps for feature identification, feature extraction, and feature selection strategies used in the chapter. Substantial analysis is conducted to empirically evaluate a series of combinations of the identified features under three feature selection scenarios, which is described in Section 3.4. In Section 3.5, experimental results are reported and discussed to locate max-performance feature combinations in each scenario, followed by frequent pattern discovery. Finally, Section 3.7 indicates implications, discusses limitations, summarizes findings, and outlines future directions of the chapter.

3.2 Related Work

The automatic prediction of review helpfulness is majorly approached via feature engineering. Under this setting, potential helpfulness related features in the form of vectors are extracted from possible review attributes. The extracted features are then concatenated into a long vector and fed into existing classification and regression models for helpfulness prediction. Several studies perform feature selection for the extracted features to choose only partly the whole collection of features for modeling more accurate helpfulness information. The identification and extraction of features and existing feature selection strategies for feature-based helpfulness prediction are introduced in subsection 3.2.1 and 3.2.2, respectively.

3.2.1 Feature-based Helpfulness Prediction

Previous studies have curated a large body of features derived from (i) review content [141, 165, 192, 322, 152, 189], (ii) review contexts such as reviewers [48, 122], social networks among reviewers [182, 291], review metadata [349, 87], and product metadata [313, 153]. Some other less frequent contextual features include review photos [320, 137], manager responses [155], travel distances [160], to name a few. Chapter 2 has given a comprehensive literature review and summarized the functionality oriented hierarchy of existing features used for helpfulness prediction. This chapter focuses on content-based features due to the ubiquitous and large use in prior literature and the ability of review texts to generalize across online platforms.

Feature-based helpfulness prediction is advantageous due to ease of implementation and interpretation. Currently, existing literature tends to identify a list of (groups of) features, homogeneous and/or heterogeneous, as inputs of a function for helpfulness prediction. The prediction performance is usually evaluated using (partial of) the whole collection of the identified features. The rationale behind using multiple features lies in attempting to better describe helpfulness characteristics.

Nonetheless, the research within domain is often fragmented and heterogeneous, posing challenges to the objective comparison, result reproduction, and findings synthetization. First, the features identified in prior research frequently lacks justification behind particular feature selection. For example, the explanation of one feature being chosen in favour of another one alike is absent, which leads to potential biases in result interpretation. Second, the categorization of features differs among the studies. So far, there is no unanimous naming and organization protocol for of the identified features. The rationale for organizing the identified features is also unclear, which impacts finding generalizability. Third, most of the existing studies evaluate helpfulness prediction models on ad-hoc datasets and the extracted features are publicly unavailable. Also, the implementation details (e.g., pre- and post- processing steps) for the task are insufficient to reproduce the experiments. The tools used for feature extraction provide various parameters and types of outputs (e.g., sentiment), which are not explicitly and clearly mentioned either. As a consequence, existing studies often report mixed findings and even contrasting results [117, 122] towards helpfulness prediction. [119] examines a list of review and reviewer related determinants in composing review helpfulness. The determinants include (1) the depth, readability, linear and quadratic rating, and age

of reviews and (2) the information disclosure and expertise of reviewers. Such determinants are found to have inconsistent behaviors in affecting the perception of review helpfulness.

The current feature-based paradigm also raises a critical question–The necessity of using all identified features for helpfulness prediction. Although extracted with different motivations, the identified features are rarely mutually exclusive. In practice, parts of the identified features may, to a certain degree, be homogeneous and share similar representations. For example, the authors in [315] show via a bivariate correlation test that several readability scores are highly correlated. Such phenomenon is called the multicollinearity issue [3], which is more likely to occur as the whole collection of features grows. The issue refers to one of the identified features being linearly predicted from the others with a substantial degree of accuracy. Quadratic and other non-linear relationships can also exist. For instance, [231] confirms that the sadness variable in the LIWC belongs to the broader negative emotion variable and the latter the affective process variable. The excessive use of features will pass redundant and/or inaccurate information into subsequent steps, and thus degrade the performance of the helpfulness prediction task. As proven in [318, 5, 199], combining all identified features or adding new features does not necessarily lead to maximum performance or improvement. Although several studies investigate the relationship among features and between features and the corresponding helpfulness, the tactics for handling feature redundancy and inter-correlation is largely understudied and ignored by existing studies.

Given the limitations identified, this chapter (1) provides the most comprehensive and generalizable survey on existing content-based features, (2) proposes a functionality oriented hierarchy scheme for organizing the identified features, (3) conducts the empirical validation of the most effective feature selection on large-scale publicly available datasets in an objective manner, and (4) releases the datasets and source code describing the implementation details used in this chapter for research reproducibility.

To the best of the author's knowledge, this chapter is the first to address the reproducibility and transferability issue of review helpfulness prediction, as well as the first work that provides the justification-driven feature selection process regardless of the platform and domain of applications. The complete and systematic literature review proves practically infeasible given largely fragmented state of the research in helpfulness prediction domain. Still, the chapter has made best efforts to report the latest state-of-art and identify the gaps to fill with the current work.

3.2.2 Feature Selection Strategies

Most existing studies investigate the predictive power of using all identified features. As discussed, the identified features often possess multi-collinearity that can decrease model performance and confound prediction results. As such, feature selection strategies are applied to the whole feature collection for helpfulness prediction prior to model construction. According to [122], an ideal feature subset is supposed to have features highly correlated with the dependent variable (helpfulness in this context) but uncorrelated with one another. Recent studies regarding feature selection for helpfulness prediction are summarized as follows.

The majority of helpfulness analysis involving feature selection remains examining limited feature subsets. In general, helpfulness related features are first identified and categorized into groups. Subsets of the features or feature groups are then manually selected to reevaluate the prediction performance. [50] proposes three groups (i.e., Unigrams, Word Embeddings, and Linguistic) of features. The following baselines are evaluated: (1) individual feature groups, (2) all feature groups combined, and (3) The combination of Unigrams and Word Embeddings, all with and without the enhancement of star ratings. [38] constructs three groups (i.e., Basic, Stylistic, and Semantic) of characteristics and predict review helpfulness using (1) individual characteristic groups, (2) the combination of both Basic and Stylistic characteristics, and (3) all groups of characteristics. [225] designs four groups (i.e., Reputation, Social, Sentiment, and Content) of features and three generic features for two datasets. The overall performance using all feature groups is reported. The performance of each feature group is also examined. [232] examines the performance of using argumentation features and bag-of-words features, both individually and in combination. Following [179], Zhang et al. [337] extract various features and divide them into five groups: Linguistic, Information Quality, Information Theory, Reviewer, and Metadata. Two types of feature combinations are employed, namely, the full set of features and leave-one-group-out subsets.

Yang et al. [322] evaluate the impact of review structure, unigrams, and three sentiment features: Geneva Affect Label Coder, Linguistic Inquiry and Word Count, and General Inquirer. The evaluation of the features is conducted individually, in combination, and as a whole. The latter two features not only improved the prediction performance, but also provided a useful interpretation to what makes a review helpful. Haque et al. [110] design three groups (Lexical, Structural, and Semantic) of features and re-

port the prediction performance using each group and all groups combined. Vo et al. [299] investigate three feature groups (i.e., Anatomical, Metadata, and Lexical) and two added features, which include (i) the number of helpfulness votes and (ii) the number of positive and negative words. The impact of the three feature groups combined is evaluated. Either and both of the added features are included in the initial feature groups for further evaluation. Kim et al. [141] investigated the effect of ten features spanning five categories (i.e., Lexical, Structural, Semantic, Syntactic and Metadata), and their combinations on helpfulness prediction. The combination of several features is tested and the one having the highest performance is highlighted. The authors found out that the most useful features were review length, unigrams, and product ratings.

[189] proposes eight sets (i.e., Visibility, Product, Reviewer, Readability, Linguistic, Sentiment, Positive Emotions, and Negative Emotions) of features. Models are trained on (1) individual feature sets and (2) the combination of the first six sets and either positive emotions or negative emotions. [318] analyzes the helpfulness of peer reviews using generic linguistic features used by [141] and manually coded specialized features. The authors first investigate the predictive power of each generic feature type in isolation. On top of that a baseline for modeling peer-review helpfulness is built by examining various combinations of the feature types. The baseline is then gradually modified by replacing generic features with and adding specialized features for prediction. [199] employs three features: unigrams, dependency bigrams, and explicit connectives, for helpfulness prediction. The predictive power of individual features and unigrams in conjunction with either dependency bigrams or explicit connectives are evaluated. Empirical results on five given datasets show unigrams achieve slightly higher accuracy among individual features. Combining either of the two features with unigrams does not lead to significant improvement. [73] proposes seven classes (i.e., Temporal, Rating, Size, Topical, Sentiment, Readability, and Content) of features. To gauge the influence of rating information within the sentiment category, the authors prepare two variants of the sentiment category, and combine the variants with all other categories, separately.

[127] realizes three categories (i.e., Content, Sentiment, and Quality) of features according to interviewees' opinions via semi-structured interviews. The content category contains four methods for specifying content-based features, whereas the remaining categories contain a set of features. The authors first search for the best method in encoding content by evaluating the combination of the Content category using different encoding methods with the Sentiment and Quality category. Subsequently, the performance of

four feature combinations are tested: (1) Content alone, (2) Content+Sentiment, (3) Content+Quality, and (4) Quality+Sentiment. [152] designs four groups (i.e., Linguistic, Metadata, Readability, and Subjectivity) of features and analyzes the performance of the predictive model when using feature groups individually and as a whole. The best predictive results are observed when the feature groups are combined. Saumya et al. [266] propose 15 basic and two additional features named product description similarity and customer question-answer similarity. The impact of the latter two features are evaluated along with the basic features in three scenarios: (1) no additional features, (2) either of the additional features, and (3) both of the additional features. [120] predicts and ranks review helpfulness, using three groups (i.e., Needs Fulfillment, Information Credibility, and Mainstreaming-opinion Divergence) of features. Further improvement is found when the features are jointly used with those proposed by [172, 141].

[97] introduces a series of features organized into a two-tier hierarchy. The hierarchy contains four categories (i.e., Product and Sales Data, Individual Review Data, Reviewer Characteristics, and Textual Analysis of Reviews), some of which further contains sub-categories of features. The authors evaluate model performance when using all available features and all possible combinations of three broad feature (sub)categories. [122] combines four proposed categories (i.e., Content, Sentiment, Author, and Visibility) of features. Four combination patterns are considered: (1) Content+Sentiment, (2) Content+Sentiment+Author, (3) Content+Sentiment+Visibility, and (4) all feature categories. [291] integrates four types (i.e., Author, Rater, Connection, and Reference) of social context features into helpfulness prediction relying on content-based features. To investigate the necessity of exploiting all introduced contextual information, each (all) of the social context features is removed. Empirical results show performance degrading when each type of social context is eliminated. The worst performance is received when all social context features are excluded. Chen et al. [47] adopted four groups (i.e., Surface, Unigram, Part-of-speech, Word Embedding) of features. The authors test model performance using (1) single feature groups and (2) a subset of features and feature groups. [171] identifies four groups of features from existing studies [322, 141, 319, 192], each as a baseline. The performance of argument-based features alone and when being used with the baseline features is evaluated.

Several studies also measure the importance of the identified feature during the prediction of review helpfulness. [107] proposes three groups (i.e., Linguistic, LCM, and Visibility) of features. The authors apply Random Forest (RF) to evaluate the combina-

tion of all groups and each one. As a by-product, the importance of individual features of the Linguistic group is also calculated. Akbarabadi et al. [5] focus on 12 features grouped into four categories (i.e., Summary, Reviewer, Text, and Readability) from reviews. The authors first report the overall performance using all categories, along with the importance of individual features in predicting helpfulness. The predictive power of each category is also reported. To show the relative importance of individual features, [225] ranks the top nine features for both datasets using information gain. [279] conducts gradient boosting decision tree (GBDT) on reviews and ranks the identified features influencing review helpfulness using the learned importance. The authors find that the average product review rating and several proposed textual features such as readability, polarity, and entropy are the most important parameters for helpfulness. Similarly, Meng et al. [197] obtained feature importance via GBDT by computing the average of the relative importance of the features over all of the trained trees. [320] designs six heuristic features in the online hotel context to model review helpfulness. Three studies are conducted via conjoint analysis to investigate (1) the importance of individual features, (2) the importance of each possible value within each of the six features, and (3) the different of the importance of the five remaining between positive (1-star and 2-star) and negative (4-star and 5-star) hotel reviews. [162] proposes 11 determinants for helpfulness evaluation. The importance of each determinant is measured via t-test and logistic regression analysis from helpfulness classification. As a result, five significant determinants encompassing product data, review characteristics, and textual characteristics of online reviews are recognized.

Several studies perform feature selection using wrapper methods. [72] captures the importance of features to review helpfulness using the Boruta algorithm [154], which is a wrapper-based measurement built upon z-score. [331] introduces "the degree of detail" feature as a function of review length and n-grams, alongside seven other features. In addition to the overall performance using all proposed features, the authors compare the importance of the features by reporting the result of all-minus-one feature combinations. The "the degree of detail" feature proves to be the most important in helpfulness prediction, leading to a significant drop in accuracy after its exclusion. [166] performs feature selection on ten stylistic features learned from review texts. The authors first train a logistic regression model to classify the binary helpfulness of reviews. The trained coefficients indicates the relevance of each stylistic of a review feature to its helpfulness. The selection is done using the backward stepwise method based on the probability of the Wald statistic to remove "insignificant" features. The performance of all features

combined is then compared against that using only the significant ones. [172] proposed a classification framework for detecting low-quality reviews. In the work, a set of features are proposed and grouped into three categories (i.e., Informativeness, Readability, and Subjectiveness). To evaluate the effectiveness of the features, the authors start with a subset of features in the Informativeness category, and incrementally add other features in the Informativeness category and those in the remaining two categories. The authors also estimate the predictive power of individual features. [125] removes variables that are not statistically significant using backward selection techniques. To this end, a regression model is trained on all available variables. The most insignificant variable causing the least decrease in R-squared statistical measure is removed one at a time. The removal process continues until all remaining variables are significant. In [337], PCA and Recursive Feature Elimination (RFE) are used to select the most informative and effective features to represent helpfulness. Both schemes show that 10 of the original 22 extracted features already reaches relatively stable performance. Similarly, [197] the RFE algorithm is used to locate the max-performance feature set prior to model construction. [21] first employs the RF classifier to rank the extracted features. RFE is then conducted to remove insignificant features using the learned importance weights. After the removal, the total number of features are reduced from $2,205$ to $1,789$.

Finally, feature selection can also be approached by dimension reduction techniques. [187] applies a chi-square test to the standard TFIDF unigram representations of reviews to select the $1,000$ most significant words. [226] extracts 17 attributes using a data quality framework [307] developed from the end user's perspective, and further groups the extracted features into five categories. The authors then examine the extent to which each category is related to review helpfulness, measured by the correlation between the five factors and helpfulness. [341] measures feature importance using Pearson correlation coefficients. Negatively correlated features are thus distinguished from positively correlated features to obtain the final feature subset for helpful review identification. [231] identifies 11 factors determining review helpfulness and explores their effect over multiple products. Pearson correlation analysis is used to check the linear relationship between the factors and the corresponding helpfulness. [179] proposes three schemes for selecting features for helpfulness prediction. The first scheme applies Principal component analysis (PCA) on three variants of the original feature matrix. The second scheme ranks the extracted features via the similarity between each feature and the corresponding helpfulness, utilizing cosine similarity, Jaccard similarity and matching similarity. The third scheme estimates the mutual information between features and the correspond-

ing helpfulness. [220, 221] examine the effectiveness of review semantics in predicting review helpfulness. Both papers employ a series of BOW models to encode semantic information. [122] performs Correlation-based Feature Selection (CFS) [106] prior to model construction. The authors report a drop from 25 to on average 9.6 in the number of features after applying greedy step-wise CFS, with only slight decrease in model performance. [220] includes the whole collection of review words, whereas [221] uses words and phrases (human concept scripts) manually highlighted by participants. The performance of the models are further compared against their dimension reduced counterparts using CFS. [22] compares model performance using all available features with that using a subset of features. The subset is obtained by applying CFS to the discrete features using the hill climbing search method.

As presented above, numerous types of analysis tasks have been conducted to select useful features for helpfulness prediction. Most of the aforementioned studies conduct feature selection in a primitive manner, lacking rationale and systematic investigation for the selected features and feature groups. For example, the predictive power of individual features and those within each category are insufficiently examined. Although several studies compute the comparative importance of features, the necessity of using excessive features and effective methods of selecting features are yet to be addressed. Existing dimension reduction techniques allows for more effective selection for feature subsets by sacrificing additional computational resources and less straightforward result interpretation. The goal of this chapter is to build prediction models using parsimonious features while preserving most model performance and robustness. To balance the trade-off between result interpretation and model efficiency, this chapter opts for wrapper methods for feature selection. Multiple feature selection scenarios are taken into account to examine model performance. The models are built upon individual features, combinations of category-level features in the identified hierarchy.

3.3 Feature Selection Computational Framework

Figure 3.1 summarizes the computational framework that conducts fEAture SelectIon for hElpfulness pRediction (EASIER). In brief, EASIER entails three steps to perform feature-based helpfulness prediction. To start with, the procedure and criteria are described to collect recent relevant literature, from which frequently cited content-based

feature candidates are identified. Each of the identified feature candidates is then introduced and the feature construction process is specified. Finally, the evaluation protocols and feature selection strategies are provided to locate max-performance feature combinations for review, followed by result analysis and discussion. The following subsections will introduce each step in more details.

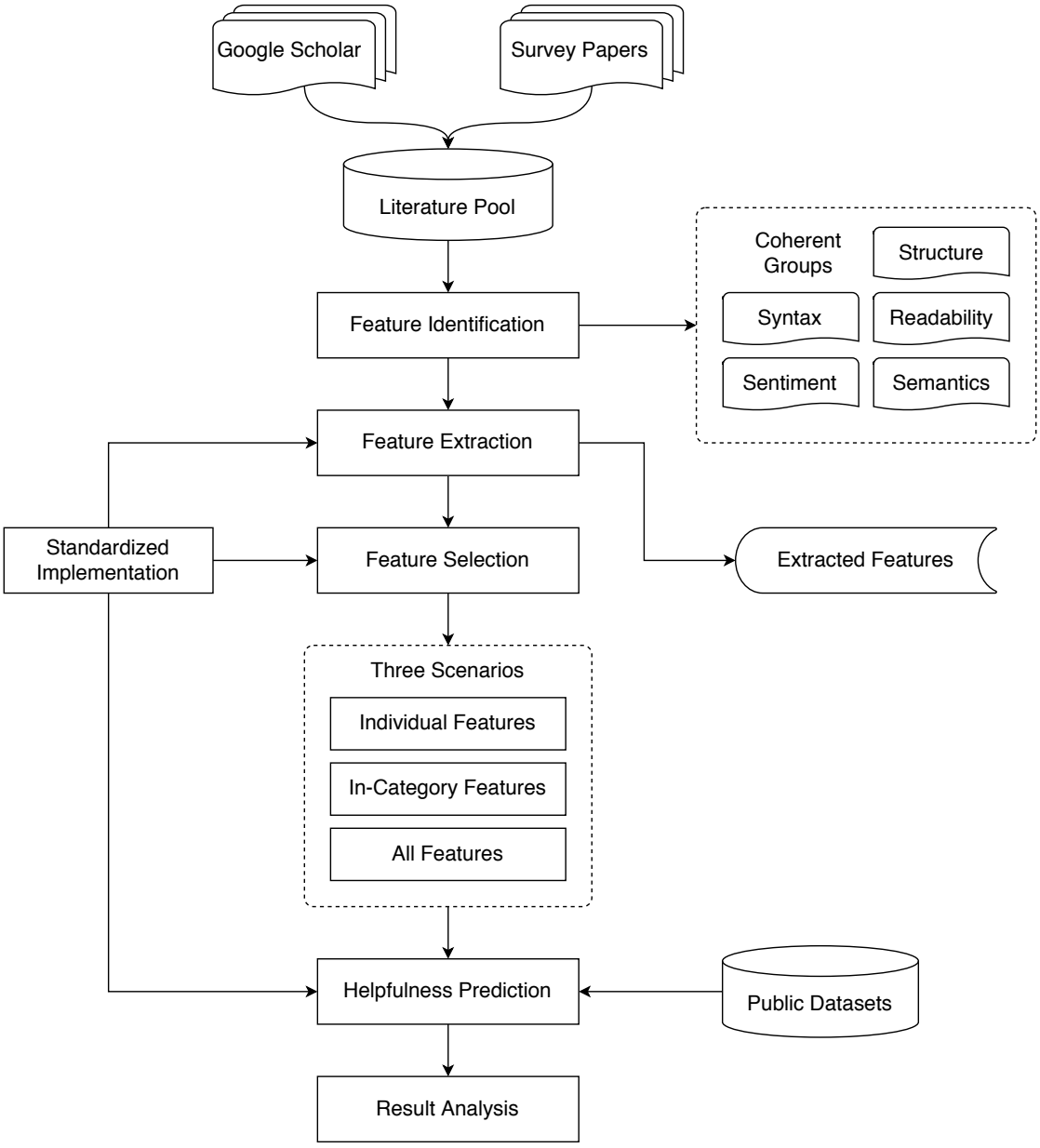

Figure 3.1: The EASIER framework.

3.3.1 Feature Identification

The chapter identifies frequently cited feature candidates from recent literature to provide wide generalization and fair comparison with the majority of studies on the topic. To this end, a collection of most recent relevant studies are first collected and filtered, from which feature candidates are identified.

Paper Acquisition The collection of relevant papers is based on (i) the references of the three most recent survey papers from the review helpfulness field [224, 117, 41] and (ii) the top 50 relevant studies retrieved from the Google Scholar database and published before 2019, using the following search query:

> ("*online reviews*" OR "*product reviews*" OR "*user review*" OR "*customer review*" OR "*consumer reviews*") AND ("*useful*" OR "*helpful*" OR "*usefulness*" OR "*helpfulness*").

Given the scope of the chapter, the 149 collected papers are filtered based on the following criteria: (i) *automated* prediction of online product review helpfulness; (ii) inclusion of *factors* influencing review helpfulness; and (iii) *English-written* reviews analysis only. As a result, 74 papers (See the "Literature" column in Table 3.1) are identified.

Feature Acquisition Features mentioned in the 74 identified papers are collected, along with the frequency of feature mentions. The following rules are adopted for feature list compilation: (i) features mentioned at least three times over the entire paper collection to exclude rare features, (ii) removal of human-annotated features due to expensive manual annotation process, and (iii) inclusion of only *content-based* features to support platform-independent generalizability and transferability. As a results, 27 feature candidates are identified.

As a novelty, the chapter additionally incorporates two semantic features and one sentiment feature that are gaining more recent attention. Such features have been proved robust in numerous text mining and natural language processing applications but are so far under-studied in review helpfulness prediction.

Table 3.1 presents the 30 content-based features identified from recent literature. The features are further grouped into five coherent categories (i.e., semantics, sentiment, readability, structure, and syntax) following the convention in the research field.

Note that context-based features such as reviewer characteristics are currently excluded from the feature pool since they are domain- and/or platform-dependent, and thus not always available.

3.3.2 Feature Extraction

The description and construction process of the identified features in groups is presented as follows. It is worth noting that some features overlap functionally, for instance, all sentiment features compute the emotional composition of reviews via different lexicons. Some features are constituents of others, such as readability scores resulting from different linear transformations of certain structural features. Following the convention in the research field, features in both cases are treated as individual ones.

Semantics

Semantic features refer to the meaning of words and topical concepts from the review content by modelling terms statistics into vectors. The five semantic features for the helpfulness prediction task are as follows:

UGR and BGR The unigram bag-of-words representation of a review uses the term frequency-inverse document frequency (TF-IDF) weighting scheme [264], where each element of a vector corresponds to a word in the vocabulary. Similarly, the bigram bag-of-words representation encodes all possible word pairs formed from neighboring words in a corpus. Both UGR and BGR ignore terms that have a document frequency value below 10 when building the vocabulary. The vector representations are then transformed into unit vectors via the L2 normalization.

LDA Latent Dirichlet Allocation representation learns the topic distribution of a review. Topic modeling considers corpus as a mixture of topics, and each topic consists of a set of words. In the case of online product reviews, the topics can be different product properties, emotional expressions, etc. The original LDA algorithm [30] is adopted to learn the probability distribution of latent topics for each review. Following [321], the number of topics is set to 100 during training.

SGNS and GV As a novelty, the chapter also uses the two most recent types of *word embeddings* as features. The Skip-Gram with Negative Sampling [201] and

Table 3.1: Features used in the analysis.

Category	Feature	Dim.[1]	Description	Literature
Semantics	UGR	V	Unigram TF-IDF	[322, 182, 141, 339, 192, 331, 171, 321, 220, 221, 187, 127, 340, 343, 73, 110, 177, 318, 191]
	BGR	V^2	Bigram TF-IDF	[141, 331, 73, 110, 199, 318]
	LDA	100	LDA topic distribution	[38, 204, 213, 321, 127, 343, 316]
	SGNS	300	Skip-gram Negative Sampling	—
	GV	300	Global Vectors	—
Sentiment	LIWC	93	Linguistic Inquiry and Word Count dictionary	[322, 321, 325, 326, 3, 54, 132, 221, 166, 231]
	GI	182	General Inquirer	[322, 141, 171, 321, 153, 276, 343, 318]
	GALC	21	Geneva Affect Label Coder	[322, 192, 171]
	OL	3	Opinion lexicon	[172, 182, 179, 279, 340, 73]
	SWN	3	SentiWordNet	[16, 15, 279, 127, 158]
	SS	3	SentiStrength	[189, 261, 209]
	VADER	3	VADER lexicon	—
Readability	FKRE	1	Flesch–Kincaid Reading Ease score	[97, 192, 189, 153, 3, 180, 221, 279, 315, 166, 340, 343, 73, 266, 316, 158]
	FKGL	1	Flesch–Kincaid Grade level	[97, 148, 152, 189, 221, 340, 343]
	GFI	1	Gunning Fog Index	[97, 148, 152, 189, 87, 326, 54, 132, 180, 340, 73]
	SMOG	1	Simple Measure of Gobbledygook	[97, 152, 189, 340, 343, 73]
	ARI	1	Automated Readability Index	[97, 148, 152, 189, 230, 54, 180, 276, 340]
	CLI	1	Coleman–Liau Index	[97, 148, 152, 189, 325, 54, 180, 340, 349, 309]
Structure	CHAR	1	Number of characters	[97, 161, 28, 38, 229, 80, 49, 54, 221, 110]
	WORD	1	Number of words	[172, 322, 182, 212, 141, 97, 148, 339, 124, 192, 225, 331, 161, 38, 171, 226, 261, 321, 179, 87, 153, 16, 48, 230, 325, 326, 54, 125, 132, 137, 180, 221, 276, 314, 315, 127, 340, 343, 73, 110, 162, 177, 266, 318, 320, 349, 155, 316, 158, 191, 309]
	SENT	1	Number of sentences	[172, 322, 182, 141, 97, 339, 192, 161, 38, 171, 226, 321, 179, 54, 221, 127, 343, 73, 110, 177, 318, 155, 191]
	AVG	1	Average number of words per sentence	[172, 322, 182, 141, 161, 38, 171, 226, 321, 179, 54, 221, 341, 343, 73, 191]
	EXCLAM	1	Number of exclamatory sentences	[322, 141, 171, 321, 177, 318]
	INTERRO	1	Number of interrogative sentences	[322, 141, 171, 321, 110, 177, 318]
	MIS	1	Number of misspelling words	[97, 161, 279, 166, 340, 343]
Syntax	NOUN	1	Number of nouns	[182, 141, 339, 192, 189, 54, 279, 343, 266, 318, 191]
	VERB	1	Number of verbs	[182, 141, 120, 339, 192, 189, 279, 343, 177, 266, 318, 191]
	ADJ	1	Number of adjectives	[182, 141, 339, 192, 189, 179, 279, 343, 177, 266, 318, 191]
	ADV	1	Number of adverbs	[182, 141, 339, 192, 189, 179, 343, 177, 318, 191]
	COMP	1	Number of comparative sentences	[182, 339, 331, 242, 191, 231]

[1] The feature dimensionality. V indicates the vocabulary size of the training set of a corpus.

Global Vectors [237] aim at learning the distributed representations of words. Under this setting, each word is mapped into a dense vector space, where similar terms display closer spatial distance. Thus, each review can be simply converted into a vector by averaging the embeddings of its constituent words, where out-of-vocabulary words are skipped.

Sentiment

Sentiment features analyze the subjectivity, valence, and emotion status of content written by customers. Previous work [42, 160] has shown relevance between helpfulness of a review and the sentiments expressed through its words. The chapter constructs sentiment features using the seven most frequently-used lexicons. The first three lexicons are category-based, each estimating the probability of a review belonging to its predefined lexicon categories. The remaining lexicons are valence-based, each looking up the valence (i.e., positive, neutral, and negative) of words in a review where possible. The comparison among the sentiment lexicons can be found in Chapter 2

LIWC The Linguistic Inquiry and Word Count dictionary [236] classifies contemporary English words into 93 categories, including social and psychological states. The dictionary covers almost $6,400$ words, word stems, and selected emoticons.

GI General Inquirer [129] attaches syntactic, semantic, and pragmatic information to part-of-speech tagged words. It contains $11,788$ words collected from the Harvard IV-4 dictionary and Lasswell value dictionary, which are assigned to 182 specified categories.

GALC Geneva Affect Label Coder [268] recognizes 36 emotion categories of affective states commonly distinguished by 267 word stems. The Geneva Emotion Wheel model [192, 322] is followed, and the 20 of the GALC categories plus an additional dimension for non-emotional words are adopted.

OL The Opinion Lexicon [121] is widely used by researchers for opinion mining. It consists of $2,006$ positive and $4,783$ negative words, along with the misspellings, morphological variants, slang, and social media markups.

SWN SentiWordNet [14] is a lexical resource for sentiment and opinion mining. It assigns to each synset of WordNet [90] three sentiment scores: positivity, negativity, and objectivity, in terms of probability.

SS SentiStrength [293] is a tool for automatic sentiment analysis on short social web texts written in informal language, incorporating intensity dictionaries, words with non-standard spellings, emoticons, slang and idioms.

VADER As a novelty, the chapter also adopts the Valence Aware Dictionary and sEntiment Reasoner [126]. VADER is a lexicon specifically attuned for social media texts. It has $3,345$ positive and $4,172$ negative terms, and is enhanced with general heuristics for capturing sentiment intensity.

Sentiment features are built as follows. For each categorical lexicon, a sentiment feature is represented by the histogram of all its predefined categories. Take LIWC as an instance, the generated feature vector of 93 dimensions contains numeric statistics of a review corresponding to each predefined category. Similarly, a feature vector derived from GI and GALC contains 182 and 21 elements encoding information of a review towards individual predefined categories, respectively.

As for valence-based lexicons, a review is described using a three-dimensional vector: the percentage of positive, neutral, and negative sentences in a review. Given a sentence, all its words are looked up in a lexicon, and the corresponding valence values are subsequently summed up. A sentence is considered positive if the total valence is greater than zero, negative if less than zero, and neutral otherwise. During the valence lookup, VADER heuristics are applied to OL and SWN to improve the detection accuracy [253]. The heuristics does not apply to SS since the toolkit offers a similar built-in mechanism for sentiment intensity evaluation.

The aforementioned sentiment features differ one another. In category-based lexicons, the sentiment of a review is described using predefined categories, similar to an opinion is understood from different perspectives. Meanwhile, valence-based lexicons detect the polarity of review words differently. For example, the term "clean" can be positive in some lexicons but neutral in others. As a result, the same review will obtain different vector representations due to various sentiment measurement criteria. Further details of the lexicon composition, such as the predefined categories and vocabulary can be found in the corresponding literature of individual lexicon and the survey papers [330, 253].

Readability

Readability measures the ease of reading texts. As pointed out by [79], even a minor increase in readability largely improves review readership, leading to more opportunities for reviews to receive helpful votes. Thus, readability has been frequently addressed in the past papers on helpfulness prediction. In EASIER, six readability tests are used to construct the readability features, taking advantage of the number of characters, syllables, words, complex words, and sentences. The readability features are calculated according to the readability test formulas, which are shown in Chapter 2 The work by Benjamin et al. [25] and original research papers provide more information regarding the motivation and explanation of the readability tests.

Similar to the sentiment category, the six readability features used in the chapter will obtain different vector representations. While referring to the same underlying concept (ease of readiness), the use of different formulas, namely linear transformations of the counting statistics, reflects different focuses on understanding the readability of a review. Since the readability tests may return values of different range, the obtained features are normalized via z-score.

Structure

Structural features count the length and occurrence of specific language unit types. The following six features are selected to represent the structure of a review. The first three features are self-explanatory, including the number of characters (CHAR), tokens (WORD), and sentences (SENT). Similarly to Xiong et al. [319], the percentage of exclamatory (EXCLAM) and interrogative (INTERRO) sentences is taken into account. Finally, the number of misspelling words (MIS) in a review is considered.

Syntax

Syntactic features consider specific types and patterns of parts-of-speech within the review content. The percentage of the most prevalent open-class word categories, namely nouns (NOUN), adjectives (ADJ), verbs (VERB), and adverbs (ADV) is estimated. Additionally, the percentage of comparative sentences (COMP) is calculated. The procedure for comparative sentence detection follows the work by Jindal et al. [130], which

employs a list of keywords and patterns to match the review sentences. Given that comparisons can take place implicitly, only explicit expressions are captured.

3.3.3 Feature Selection

Feature-based helpfulness prediction is formulated as a binary classification (either helpful or unhelpful) problem. Most existing studies approach the task either by classification or regression. This chapter adopts the former due to its intuitive and simple output to customers.

The task of feature-based helpfulness prediction is formally defined as follows. Let $\mathcal{D} = \{(D_1, u_1), \ldots, (D_n, u_n)\}$ be a collection of n product reviews, where D is the content of a review and u the accompanying helpfulness information ($u = 1$ helpful and $u = 0$ unhelpful). Each review content $D \in \mathcal{D}$ is associated with a set of features, denoted by $\mathcal{F}(D) = \{f_1(D), \ldots, f_m(D)\}$, via m different feature extractors $\{f\}$. The goal of the task is to train a binary classifier C that searches for the max-performance feature combination $\hat{\mathcal{F}}$ from the feature pool $\mathcal{F}$ to approximate the helpfulness u such that:

$$\hat{\mathcal{F}} = \arg\max_{\mathcal{F}' \subseteq \mathcal{F}(D)} \sum_{D \in \mathcal{D}} \mathbb{1}(u = C(\mathcal{F}')), \tag{3.1}$$

where $\mathbb{1}(\cdot)$ is an indicator function.

Ideally, the search of $\hat{\mathcal{F}}$ would be exhausting all $2^m - 1$ possible feature combinations. However, such strategy requires tremendous computational resources throughout the searching process and only works for a small set of identified features. Given $m = 30$ features identified in Table 3.1, the exhaustive search strategy is not suitable due to the exponential complexity of calculation.

Instead, the search is fulfilled by a wrapper method, specifically, the step forward feature selection. Given the feature pool, the search starts with the evaluation of each feature and selects the one with the highest performance. Subsequently, all possible combinations between the selected feature and each of the remaining features are evaluated, and the second feature is selected. The iteration continues until adding features cannot improve the prediction performance. As a result, the selected features together form the max-performance feature combination. Note that this chapter focuses on using parsimonious features for effective helpfulness prediction and thus opts for forward feature selection over the backward counterpart. Although redundant features do not

necessary degrade model performance, using forward selection fits the problem-solving principle called Occam's razor and benefits from both faster training speed and less computational resources.

As for the binary classifier C, linear Support Vector Machine (SVM) is chosen given its wide adoption and high performance in previous studies on the task [141, 120, 331, 152]. The SVM algorithm aims at learning a hyper-plane to best separate helpful and unhelpful reviews by maximizing its margin to the nearest data point of both classes in the feature space. The use of the most common linear SVM classifier facilitates fair comparison between the studies within the same field. In addition, linear SVM is advantageous since it scales better to large numbers of training samples, which suits the situation in this chapter. As will be discussed in Section 3.5, the last reason for employing the linear kernel over non-linear implementation is to obtain the trained model coefficients as interpretable by-products to indicate feature importance.

3.4 Experiment Settings

This section conducts substantial helpfulness prediction analysis using the 30 identified content-based features. In Section 3.4.1, the large-scale publicly available datasets and pre-processing steps used throughout the chapter are first described. Subsequently, Section 3.4.2 gives the implementation details necessary for reproducible feature extraction and model construction. Finally, Section 3.4.3 conducts correlation analysis on the extracted features to study the mutual relationship between features.

3.4.1 Datasets

The analysis is conducted on the largest publicly available Amazon 5-core dataset [113]. Amazon is the largest Internet retailer, which has accumulated large-scale user-generated reviews. The helpfulness of such reviews is rated by online customers, which makes it an ideal candidate for review helpfulness prediction task. In fact, Amazon product reviews are predominantly used and analyzed in previous studies. Thus, adopting Amazon reviews allows for fair comparison with previous studies. Also, the analysis results can hopefully provide practical insights into the context of online business and user-generated content quality evaluation.

The original dataset consists of 24 domains, covering 142.8 million reviews collected between May 1996 and July 2014. The six domains with the highest number of reviews are selected for the chapter. The domains include Apps for Android, Video Games, Electronics, CDs and Vinyl, Movies and TV, and Books. For simplicity, the first domain is called D1, the second D2, and so on. Table 3.2 presents a review sample randomly select from the domain of Video Games, along with the accompanying attributes. As depicted in the table, each review contains a set of attributes, including (1) the ID of the targeted product, (2) the helpfulness information, namely the number of helpful and unhelpful votes given by online customers, (3) the star rating of the review, (4) the published date, week, and time of the review, (5) the ID and name of the reviewer, and (6) the summary headline and review text commenting in detail on the product. This chapter focuses on extracting content-based features from review texts, which refer to the concatenation of review summary and review text. Table 4.1 further presents the helpful versus unhelpful review examples across the six domains.

Table 3.2: Example Amazon review composition. Typos and grammatical errors are intentionally preserved.

Attribute	Value
Product ID	9625990674
Total number of votes	17
Number of helpful votes	15
Star Rating	4
Review Time	Thursday, January 19, 2012 12:00:00 AM
Reviewer ID	A16SAFL1YSO4HJ
Reviewer Name	NRage224
Review Summary	Xbox 360 Controller Skin, Black Silicone
Review Text	This is not the first one of these I have had, in fact this is about my 8th. There are many different skin types and different skin makers, so you have to judge each on it's own merit. My controller skin arrived today, well ahead of expected delivery, just as described solid black and silicone. Fit is just perfect, and installation had no [...]

Similar to tweets and other user-generated content [174], online reviews tend to be short, less ungrammatical, and using more informal expressions (e.g., neologisms, Internet slang and abbreviations), which introduce noise into the dataset. To improve data quality, the following pre-processing steps are applied to the raw online reviews. (1)

This chapter focuses on English review helpfulness prediction. For this purpose, blank and non-English reviews are filtered out. (2) Identical and nearly identical reviews [68] are common on Amazon. To avoid training data redundancy, only the ones with the highest number of votes are retained. Following the definition given by [141], two reviews are nearly identical if more than 80 percent of their bigram occurrence is shared. (3) Several reviews only have few votes and the prediction of which may lead to biases that only reflect a small group of people' attitudes towards review helpfulness. To alleviate the effect of words of few mouths [258, 334], reviews with less than 10 votes are skipped. (4) Each of the remaining reviews is lowercased and tokenized in to a sequence of words. (5) Minimum stopword removal is applied by eliminating articles (i.e., a, an, and the) from the reviews. Further stopword removal is not considered since some stopwords can be useful in building review helpfulness. For example, negation expressions such as "not" and "never" are often considered as stopwords, which flip the opinions written by customers.

The pre-processed reviews are then labeled and split prior to model training. The helpfulness label of the pre-processed reviews is determined in an automatic manner using existing human assessment provided by online users. One standard method is based on the ratio of helpful votes, which is the number of helpful votes divided by the total number of votes. Subsequently, the continuous ratio is then converted into binary helpfulness labels via a pre-defined threshold. This chapter sets the threshold to 0.6, which is the most commonly used threshold in prior research [97, 152, 189]. For each domain, a review is labeled as helpful if its ratio of helpful votes is equal to or more than 0.6, indicating that at least 60% of users believe the review is helpful. Otherwise, reviews with ratio less than 0.6 are labelled as unhelpful. To avoid the class imbalance problem, which is outside the scope of this chapter, helpful reviews are randomly sampled to have the same number as unhelpful ones and vice versa. On the other hand, reviews in each domain are partitioned using a unified scheme. Specifically, stratified random split is adopted: the whole collection of reviews is first shuffled (with a fixed seed), and then 80%, 10%, and 10% of the reviews are randomly selected respectively to build the training set, validation set, and testing set. During the selection, the percentage of samples for each class is preserved. Throughout this chapter, all feature combinations are trained on the training set, compared and selected on the validation set, and evaluated on the test set serving as unseen data in reality. The validation set is also used for tuning training parameters for the SVM classifier.

Table 3.3 demonstrates the simple descriptive statistics of the six domains sorted by data size in ascending order. From D3 to D6, while having different number of reviews, the domains approximately have 16 words per sentence and 14 sentences per review, composing about 240 words per review. D1 has comparably shorter reviews, with only one-fifth (one-third) of the words (sentences) per review. The short nature of reviews in D1 also leads to the highest OOV rate in both validation and test set, which is about twice as much as other domains. On the contrary, D2 tend to have longer reviews, with about 90 (5) additional words (sentences) more than the four domains. More descriptive statistics and discussions of the six domains can be found in Appendix 7.

Table 3.3: Descriptive statistics of the balanced domains after pre-processing.

Domain		#Reviews	#Words	$\frac{\#\text{Words}}{\#\text{Reviews}}$	#Sentences	$\frac{\#\text{Sentences}}{\#\text{Reviews}}$	$\frac{\#\text{Words}}{\#\text{Sentences}}$
D1	Apps for Android	20,416	1,204,921	59.02	106,242	5.20	11.39
D2	Video Games	23,100	7,714,545	333.96	468,771	20.29	16.48
D3	Electronics	33,962	8,515,804	250.75	536,704	15.80	15.52
D4	CDs and Vinyl	105,934	23,941,259	226.00	1,461,680	13.80	16.57
D5	Movies and TV	164,052	42,152,922	256.95	2,500,454	15.24	16.72
D6	Books	306,430	74,261,016	242.34	4,384,372	14.31	16.28

The vote distributions are further presented in Figure 3.2, displaying a similar pattern for each domain that high frequency of reviews have a relatively low number of votes.

3.4.2 Implementation

In this chapter, all the analysis tasks are coded with the Python (version 3.6) programming language. The code implementations are run on Ubuntu 16.04.5 Long Term Support as the system environment. The main hardware configuration is as follows: Intel Core i5-9600K CPU @ 3.70GHz $\times$ 6, Samsung SSD 970 EVO, and 32GB RAM.

The proposed EASIER framework subsequently extracts the aforementioned features from review texts across domains. This section endeavours to give maximum coverage of the implementation details used in EASIER. The goal is to ensure reproducible helpfulness prediction, in particular, feature data extracted from online reviews and experimental results. Text pre-processing, part-of-speech tagging, and feature extraction are done using NLTK [27]. Specifically, both SGNS trained on 100 billion words from Google News and GV trained on 840 billion words from Common Crawl

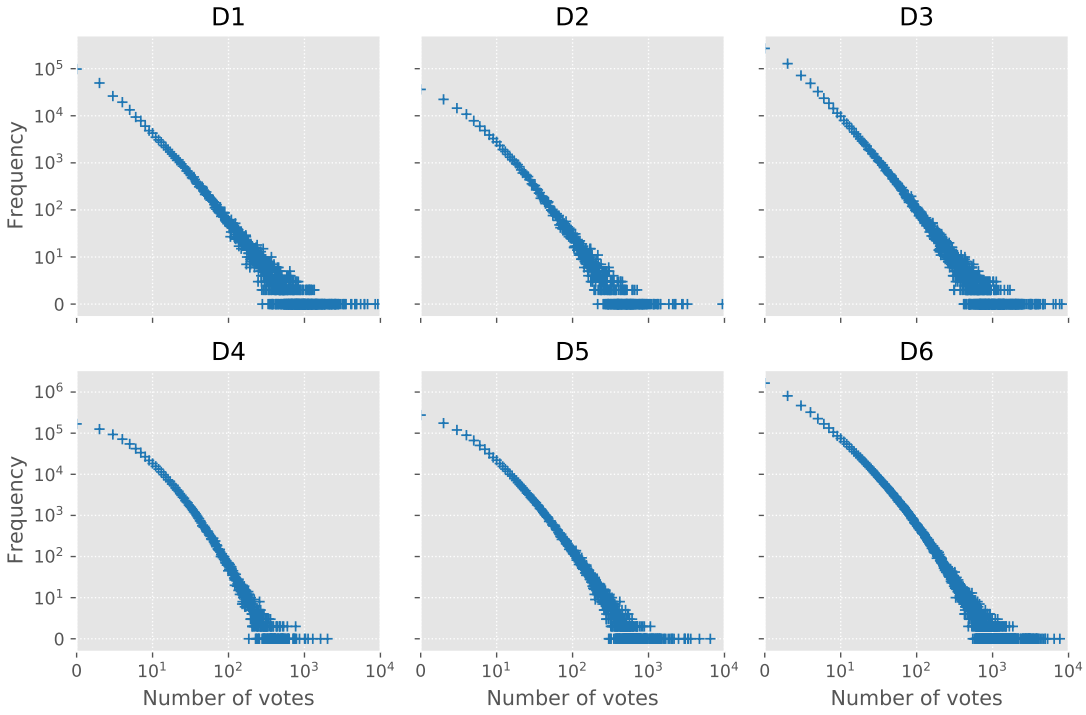

Figure 3.2: Review vote distributions.

are publicly available online. Regarding the sentiment category, LIWC 2015, the commercial version (February 2017) of SentiStrength, and VADER 3.2.1 are employed for feature extraction. The remaining lexicons are acquired as per the papers. All the readability scores are computed via the `textstat` library. As for syntactical features, the Hunspell spell checker is used to detect misspelling words. To enable the detection for product brands and contemporary language expressions, Hunspell is extended with Wikipedia titles (Retrieved February 13, 2019, from Wikimedia dump service). Finally, the TFIDF modeling, LDA topic modeling, and linear SVM classifier [297]is developed using Scikit-learn [235]. For reproducibility, all randomization processes involved in the chapter are initialized with the same random seed.

3.4.3 Feature Correlation Analysis

Prior to constructing prediction models, correlation analysis is conducted on the extracted features. As discussed, the multi-collinearity issue occurred in several features, such as in-category features sharing similar functionality and features that are partly

build upon the others. To this end, Pearson correlation tests are used to measure the mutual relationship between one feature and another. Features in the semantics category, and similarly those derived from category-based sentiment lexicons, are omitted due to the large vocabulary of unigrams and predefined lexicon categories. As for sentiment features derived from polarity-based lexicons, each one is split by its polarity dimensions into three sub-features.

As an example, Figure 3.3 demonstrates the correlation between the identified features extracted from reviews in D1. The correlation analysis of the other domains shares similar results and thus is not shown here. Overall, higher correlation is found between features within the same category. In addition, readability and structural features are highly correlated the former is constructed based on the latter. Several correlation patterns are further observed. Most negative (positive) dimensions are negatively (positively) correlated to the positive ones. In particular, strong positive correlation is found among the SS, OL, and VADER dictionaries. SMOG, EXCLAM, INTERRO, and MIS less correlated among all features. AVG acts similarly but is more correlated to several readability features as part of the readability test formulas. The number of chars, words, and sentences are found correlated to that of open-class words and comparable sentences. The reason is that longer reviews provide higher proportion of those elements. The remaining feature pairs, whether positive or negative, are weakly correlated.

3.5 Result Analysis

The empirical results are analyzed to obtain insights into feature selection for helpfulness prediction. Throughout the analysis, the performance of review helpfulness prediction is measured by classification accuracy and its ranking. The latter is provided as another prioritization measure to capture the general trend of feature performance since the accuracy of a feature (set) can largely vary in domain.

The chapter considers three scenarios for feature selection: (i) individual features (Section 3.5.1), (ii) features within each category (Section 3.5.2), and (iii) all features (Section 3.5.3). The following subsections investigate three research questions:

***RQ1**: What is the effect of individual features on review helpfulness prediction across domains?*

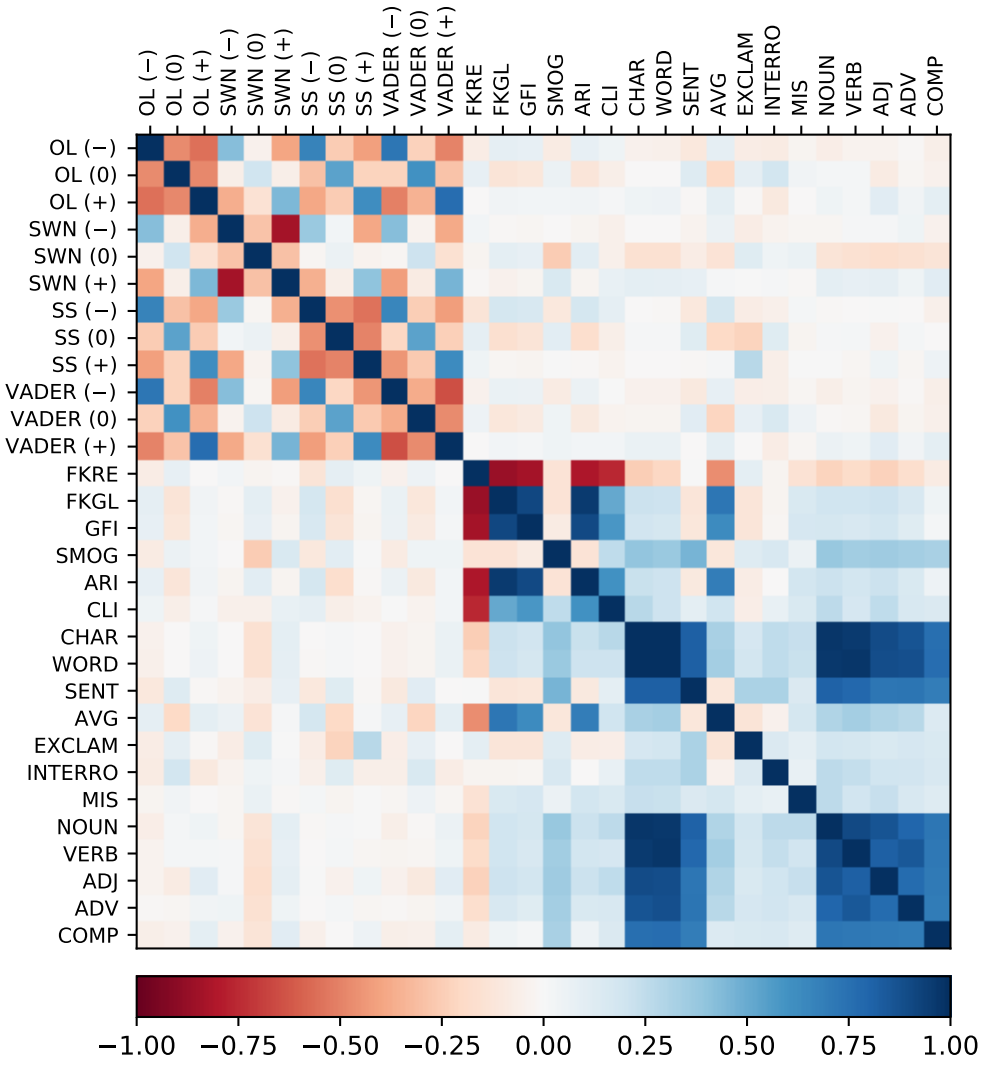

Figure 3.3: Feature correlation matrix. The −, 0, + notation respectively marks the positive, neutral, negative dimension of the sentiment features.

RQ2: *What are the max-performance combinations of features within a category for review helpfulness prediction across domains?*

RQ3: *What are the max-performance combinations of all features for review helpfulness prediction across domains?*

3.5.1 The Predictive Power of Individual Features

To answer RQ1, Table 3.4 demonstrates the classification accuracy, in-category ranking, and overall ranking of individual features, respectively. As shown, the semantics and sentiment category in general perform better than the other three categories.

Table 3.4: The classification accuracy and ranking (In-category/Overall) of individual features.

Category	Feature	D1		D2		D3	
		Accuracy	Ranking	Accuracy	Ranking	Accuracy	Ranking
Semantics	UGR	66.06	3 / 4	74.98	2 / 2	74.1	1 / 1
	BGR	60.72	4 / 10	70.74	4 / 5	69.07	4 / 5
	LDA	55.39	5 / 12	67.23	5 / 7	63.1	5 / 10
	SGNS	67.58	2 / 2	75.15	1 / 1	73.28	3 / 3
	GV	68.66	1 / 1	74.94	3 / 3	73.34	2 / 2
Sentiment	LIWC	66.16	1 / 3	73.94	1 / 4	70.78	1 / 4
	GI	63.76	2 / 5	69.18	2 / 6	67.07	2 / 6
	GALC	55.88	7 / 11	53.81	7 / 28	58.21	7 / 21
	OL	62.78	4 / 7	62.99	4 / 17	66.83	3 / 7
	SWN	61.41	5 / 8	63.12	3 / 16	62.54	5 / 12
	SS	60.77	6 / 9	58.05	6 / 22	60.86	6 / 19
	VADER	63.42	3 / 6	62.55	5 / 18	64.6	4 / 8
Readability	FKRE	52.99	4 / 23	58.44	3 / 21	55.36	5 / 26
	FKGL	51.22	6 / 28	56.84	5 / 25	55.71	4 / 25
	GFI	53.48	2 / 21	56.15	6 / 26	53.18	6 / 29
	SMOG	53.23	3 / 22	61.34	2 / 20	59.12	1 / 20
	ARI	51.37	5 / 27	57.1	4 / 24	55.83	3 / 24
	CLI	53.82	1 / 19	62.08	1 / 19	56.45	2 / 22
Structure	CHAR	54.36	3 / 16	65.24	1 / 9	62.07	1 / 13
	WORD	54.55	2 / 15	64.76	2 / 11	61.86	2 / 14
	SENT	52.69	5 / 26	63.94	3 / 15	61.8	3 / 15
	AVG	52.94	4 / 24	57.71	4 / 23	56.18	4 / 23
	EXCLAM	51.13	6 / 29	52.81	7 / 30	53.86	6 / 28
	INTERRO	55.29	1 / 13	53.16	6 / 29	51.32	7 / 30
	MIS	50.78	7 / 30	54.29	5 / 27	55.27	5 / 27
Syntax	NOUN	54.06	2 / 17	64.94	2 / 10	61.77	3 / 16
	VERB	53.72	4 / 20	64.2	4 / 13	61.09	5 / 18
	ADJ	55.14	1 / 14	65.41	1 / 8	63.27	1 / 9
	ADV	52.89	5 / 25	64.16	5 / 14	61.74	4 / 17
	COMP	53.97	3 / 18	64.33	3 / 12	62.89	2 / 11
Category	Feature	D4		D5		D6	
		Accuracy	Ranking	Accuracy	Ranking	Accuracy	Ranking
Semantics	UGR	80.83	3 / 3	78.06	1 / 1	75.02	1 / 1
	BGR	76.89	4 / 5	74.37	4 / 4	71.03	4 / 4
	LDA	65.49	5 / 11	63.39	5 / 10	61.67	5 / 9
	SGNS	81.02	2 / 2	77.26	3 / 3	73.91	3 / 3

	GV	81.06	1 / 1	77.41	2 / 2	74.04	2 / 2
Sentiment	LIWC	76.99	1 / 4	72.25	1 / 5	68.58	1 / 5
	GI	72.07	2 / 6	70.75	2 / 6	67.04	2 / 6
	GALC	56.14	7 / 27	54.86	7 / 27	55.17	7 / 23
	OL	70.13	3 / 7	66.57	3 / 7	63.64	3 / 7
	SWN	65.29	6 / 13	63.87	5 / 9	60.92	5 / 10
	SS	65.6	5 / 10	61.23	6 / 17	60.6	6 / 11
	VADER	67.68	4 / 8	65.32	4 / 8	62.51	4 / 8
Readability	FKRE	62.38	4 / 20	59.15	5 / 24	53.64	6 / 27
	FKGL	62.12	5 / 22	59.25	4 / 23	54.92	4 / 25
	GFI	61.66	6 / 23	58.48	6 / 26	54.16	5 / 26
	SMOG	64.8	1 / 15	61.32	1 / 16	56.45	1 / 21
	ARI	63.15	3 / 19	59.46	3 / 22	55.57	2 / 22
	CLI	64.58	2 / 16	60.72	2 / 19	54.98	3 / 24
Structure	CHAR	64.85	1 / 14	62.56	1 / 13	58.79	2 / 14
	WORD	63.95	2 / 17	62.17	2 / 14	58.87	1 / 13
	SENT	61.64	3 / 24	60.51	3 / 20	57.94	3 / 18
	AVG	59.83	4 / 26	58.87	4 / 25	56.59	4 / 20
	EXCLAM	53.99	6 / 29	52.79	6 / 29	53.05	5 / 28
	INTERRO	53.43	7 / 30	53.24	5 / 28	51.8	6 / 29
	MIS	55.72	5 / 28	52.35	7 / 30	50.51	7 / 30
Syntax	NOUN	65.38	2 / 12	63	2 / 12	58.92	1 / 12
	VERB	62.27	4 / 21	60.81	4 / 18	58.41	4 / 17
	ADJ	65.63	1 / 9	63.11	1 / 11	58.65	2 / 15
	ADV	61.04	5 / 25	60.24	5 / 21	57.76	5 / 19
	COMP	63.6	3 / 18	61.56	3 / 15	58.56	3 / 16

Semantics The semantics category consists of most of the globally top-five features. The best overall performance lies in semantic features directly modeling review content, leading to more dimensions for encoding information. In particular, UGR sets a strong baseline in all domains, which indicates that specific term occurrences differ between helpful and unhelpful reviews. Both GV and SGNS show comparable or higher performance than UGR, with about 1% in accuracy lower than UGR in the worst case. The promising performance demonstrates the efficacy of traditional machine learning algorithms trained on general-purpose distributed word representations for helpfulness prediction. GV outperforms SGNS in all domains except D2, being a preferable option. In contrast, BGR scores 4%–5% lower compared with UGR, suggesting increased data sparsity while using bigram features. LDA consistently ranks the lowest within the category and is even lower than several features in the sentiment and syntax category. The inferior performance can be attributed to short product reviews hindering the training

of topic distributions, which explains the lowest (highest) overall LDA ranking on D1 (D2).

Sentiment The sentiment category shows mixed performance. As for the categorical lexicons, LIWC, GI, and GALC rank respectively first, second, and last in all domains. LIWC outperforms UGR in D1 but is beaten by other domains. The accuracy gap, ranging from 1% to 6%, is proportional to data size. As such, LIWC can substitute for semantics when applied to small datasets. While the drastic low performance of GALC results from its few predefined categories and low vocabulary coverage compared with LIWC, GI shows that having almost double the size of predefined lexicon categories and words does not necessarily bring higher performance. On the other hand, the valence-based lexicons perform variously depending on data size. In most cases, OL and VADER produce higher accuracy than SWN and SS. Starting from D3, a more precise pattern that OL>VADER>SWN>SS is observed. OL generally performs better than other valence-based lexicons because it is originally generated from Amazon reviews, and thus more related to the tested domains. The results from the category show that the predictive power of lexicon-based features highly depends on the definition of lexicon categories, vocabulary coverage, as well as data size.

Readability, Structure and Syntax Features from the remaining three categories generally have less individual predictive power. The majority of the features have lower rankings, with the accuracy about 10%-27% inferior to UGR. The low performance indicates the indistinguishable nature among classes. In the readability category, for instance, similar scores are observed regardless of helpfulness of a review. Likewise, both helpful and unhelpful reviews are characterized by similar ratio of exclamatory and interrogative sentences, as well as misspellings. As a result, such features are less preferable in the helpfulness prediction task when used individually. Still, the slightly improved accuracy in the syntax category indicates that helpfulness is more related to the proportion of open-class words. In particular, ADJ generally performs better than other syntactic features due to the descriptive nature of products or general purchase satisfaction/dissatisfaction.

To better understand the behaviour of individual features across domains, the mean and standard deviation of the overall ranking of each feature are produced in Figure 3.4. The former describes the average performance of a feature, whereas the latter describes the stability of feature performance. As demonstrated, GV, SGNS, UGR, LIWC, and GI

are the most ideal features with both excellent performance and stability. Those features show the feasibility of helpfulness prediction by modeling semantics and sentiment of product reviews. The remaining features, however, have either less satisfactory or stable performance.

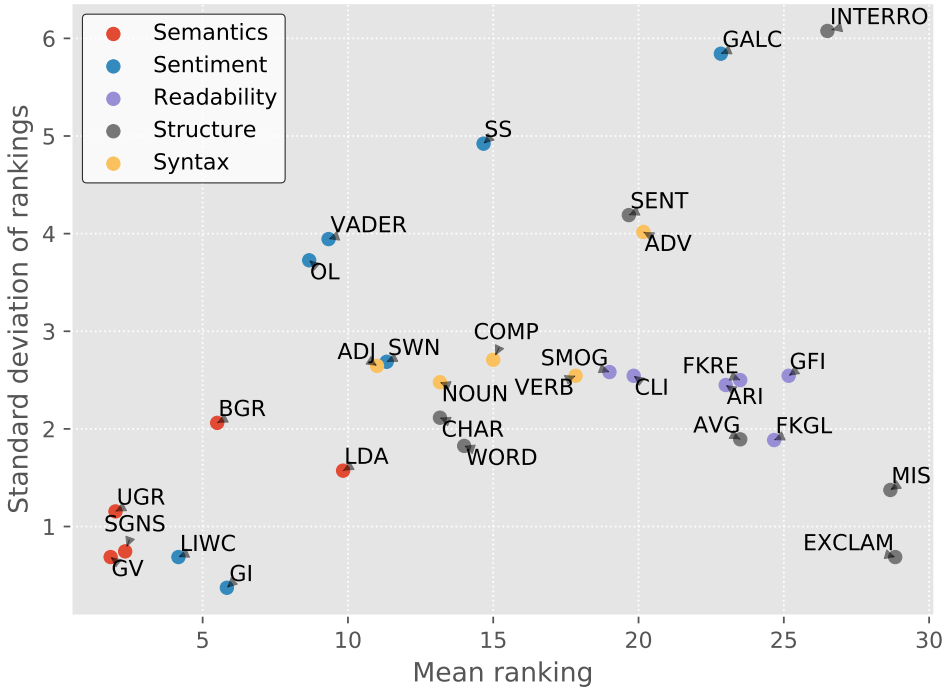

Figure 3.4: Feature performance versus stability.

3.5.2 Combinations of Features within Each Category

To answer RQ2, Table 3.5 presents the accuracy and ranking of the max-performance feature combination in each category. As shown, BGR is the only feature not being selected in any scenarios due to the associated sparsity. Also, all domains demonstrate an identical ranking of feature categories, with the semantics, sentiment, and structure category playing the dominant role in helpfulness prediction.

To evaluate the benefit of combining multiple features within the same category, the max-performance feature combination is compared with the most promising individual feature. As Figure 3.5 illustrates, in all but one cases, using multiple features achieves

Table 3.5: Max-performance combinations of features within each category.

	Category	Accuracy	Ranking	Combination
D1	Semantics	67.53	1	GV+SGNS
	Sentiment	67.29	2	LIWC+OL+GI
	Readability	54.06	5	SMOG+CLI+FKGL
	Structure	56.51	3	SENT+INTERRO+EXCLAM+AVG+WORD+CHAR
	Syntax	55.14	4	ADJ+ADV+COMP
D2	Semantics	76.67	1	SGNS+GV+LDA
	Sentiment	74.81	2	LIWC+VADER
	Readability	64.46	5	CLI+SMOG+GFI+ARI+FKGL
	Structure	66.67	3	CHAR+WORD+MIS+INTERRO
	Syntax	65.54	4	NOUN+ADJ+ADV+VERB
D3	Semantics	74.10	1	GV+SGNS
	Sentiment	73.98	2	LIWC+OL+GI+GALC+VADER
	Readability	59.12	5	SMOG
	Structure	64.10	3	CHAR+INTERRO+WORD
	Syntax	63.60	4	ADJ+ADV
D4	Semantics	81.32	1	UGR+GV
	Sentiment	78.81	2	LIWC+OL+GI+SS+SWN+VADER
	Readability	65.52	5	SMOG+ARI+FKGL
	Structure	68.68	3	CHAR+INTERRO+WORD+MIS+SENT
	Syntax	67.36	4	ADJ+ADV+VERB+NOUN
D5	Semantics	78.18	1	UGR+GV
	Sentiment	74.95	2	LIWC+GI+OL+VADER+SS+SWN
	Readability	62.05	5	SMOG+CLI+GFI+ARI+FKRE
	Structure	65.92	3	CHAR+INTERRO+WORD+EXCLAM+MIS
	Syntax	63.89	4	NOUN+VERB+ADV+COMP
D6	Semantics	75.49	1	UGR+SGNS+LDA
	Sentiment	71.45	2	LIWC+GI+OL+SWN+SS+GALC
	Readability	59.05	5	SMOG+ARI+FKRE+CLI+FKGL
	Structure	61.59	3	WORD+INTERRO+MIS+EXCLAM+SENT
	Syntax	59.11	4	NOUN+COMP+VERB+ADV

better performance on a category level. The rationale is that combining features provides new descriptive information of reviews and allows the information to complement one another. The improvement, depending on domains, tends to be more noticeable in the sentiment, readability, and structure category. On D1, GV alone reports higher accuracy than the max-performance combination GV+SGNS in the semantics category since the domain has a large proportion of OOV words. As shown in Table 3.3, D1 has the shortest average length but highest OOV rate, which is about twice as much as other domains. Further manual inspection reveals that many OOV words are domain-specific terms such as names of mobile applications and mobile games. Moreover, only

53% of the OOV words overlap between the validation and test set. When converting reviews into embeddings, the OOV issue in the pre-trained SGNS model further affects the performance, which explains why GV+SGNS is worse than GV and less robust on D1.

The average number of features within each category used for helpfulness prediction is provided in Table 3.6. Frequent feature combination patterns that occur at least four times across domains are extracted via the PrefixSpan algorithm. The constant use of LIWC, SMOG, ADV, and INTERRO+WORD is observed, and thus it is recommended to include them for max-performance feature combinations within the corresponding categories. As for the sentiment category, adding GI+OL (VADER alone) on top of LIWC can achieve higher performance in five (four) of six domains. Similarly, using INTERRO+WORD in conjunction with CHAR (MIS) can improve the structure category in five (four) domains. Furthermore, including one of ARI, CLI, and FKGL in addition to SMOG in the readability category helps to increase the accuracy in four domains. The same applies to ADJ and NOUN+VERB for ADV in syntax category. Finally, the semantics category tends to have various feature combinations, with GV and SGNS being prevalent in most cases.

Table 3.6: Frequent feature combination patterns within each category.

Category	#Features	Pattern (Frequency)
Semantics	2.33 ± 0.47	GV (5)
Sentiment	4.67 ± 1.60	LIWC (6)
		GI+OL+LIWC (5)
		LIWC+VADER (4)
Readability	3.67 ± 1.49	SMOG (6)
		ARI+SMOG (4)
		CLI+SMOG (4)
		FKGL+SMOG (4)
Structure	4.67 ± 0.94	INTERRO+WORD (6)
		CHAR+INTERRO+WORD (5)
		INTERRO+MIS+WORD (4)
Syntax	3.50 ± 0.76	ADV (6)
		ADJ+ADV (4)
		ADV+NOUN+VERB (4)

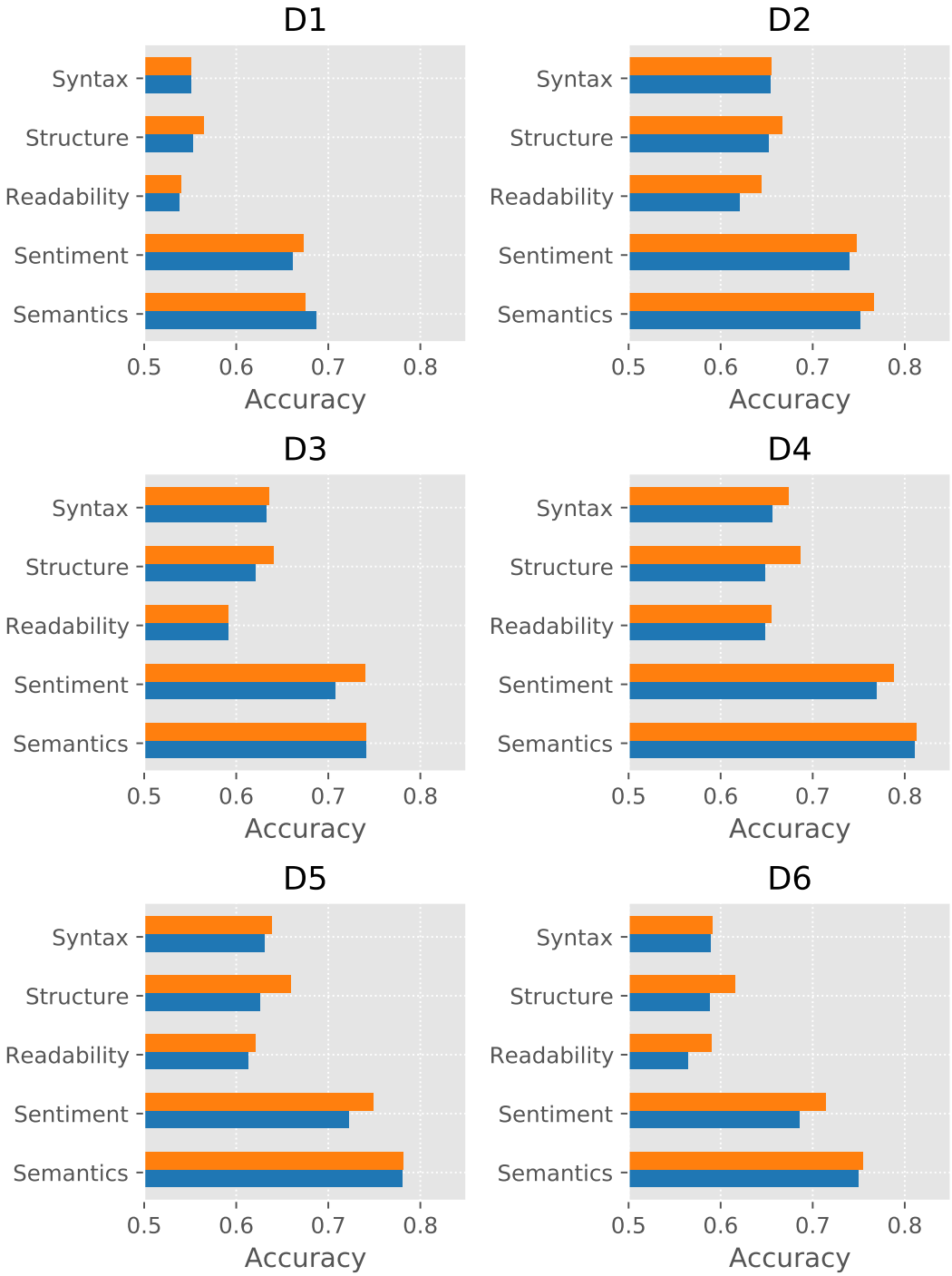

Figure 3.5: Max-performance comparison between individual features (Blue) and the combination of in-category features (Orange).

3.5.3 Combinations of All Features

To answer RQ3, the final result of review helpfulness prediction using the max-performance feature combination from all categories are presented in Table 3.7. The max-performance combinations contain four to seven features selected from only 18 out of the 30 features. Some of the 12 excluded features have excellent individual performance or are popular in category-level combinations, such as GI and WORD. The exclusion is due to features selected earlier (partly) contain information provided by those later. Despite no clear-cut patterns across domains are observed from the combinations, the semantics, sentiment, and syntax category play more important role in forming the max-performance feature combinations. Especially, GV, UGR, LIWC and ADJ are used on half of the occasions.

Table 3.7: Max-performance combinations of all features. The involved categories are listed below each feature combination.

	Accuracy	Combination
D1	69.78	GV+VADER+ADV+SWN Semantics, Sentiment, Syntax
D2	76.80	SGNS+GV+LDA+MIS+INTERRO+ADJ Semantics, Structure, Syntax
D3	75.96	GV+SGNS+LIWC+OL+GALC+ARI Semantics, Sentiment, Readability
D4	83.09	UGR+LIWC+ARI+INTERRO+CLI+ADJ+VERB Semantics, Sentiment, Readability, Structure, Syntax
D5	79.72	UGR+CHAR+LIWC+ADJ Semantics, Sentiment, Structure, Syntax
D6	76.20	UGR+SMOG+ADV+VADER Semantics, Sentiment, Readability, Syntax

The accuracy among the max-performance individual feature, feature combination within each category, and combination of all features is further compared in Figure 3.6. As shown, using features from multiple categories consistently achieves the highest performance. Similar to using multiple features within a category, the improvement lies in features from different categories together describe a review from multiple perspectives, making the vector representations more comprehensive.

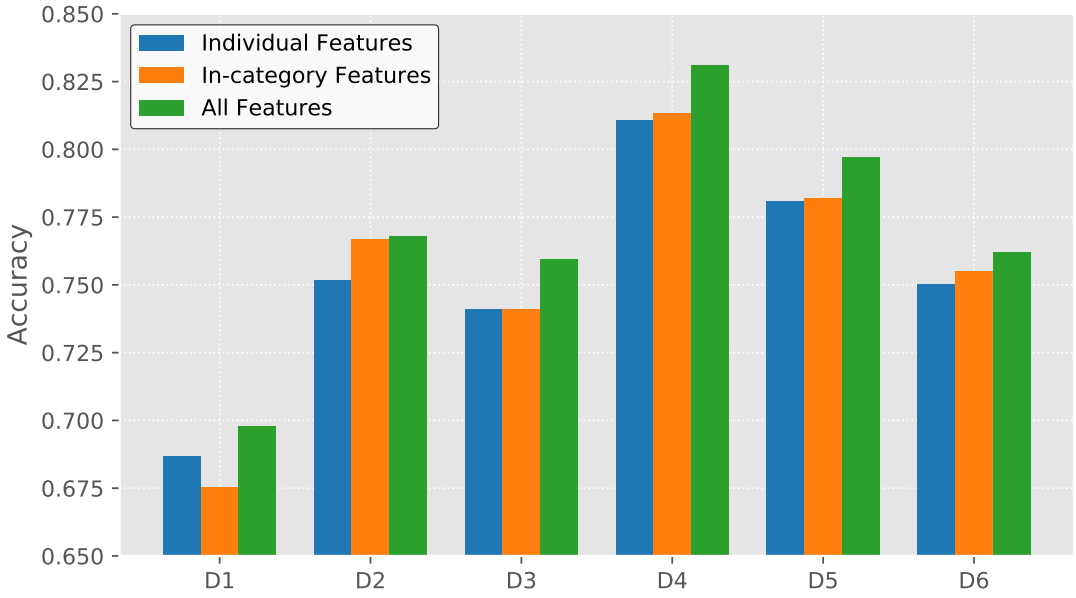

Figure 3.6: Max-performance comparison among individual features (Blue), the combination of in-category features (Orange), and the combination of all features (Green).

3.6 Discussions

The last three subsections have answered the research questions via a series of max-performance feature combinations across domains, along with their predictive power. Extensive analysis shows that appropriately increasing the number of features can increase the performance of helpfulness prediction in almost all cases, regardless of feature categories and feature selection scenarios. As discussed, those performance gains lie in multiple features helping model a review's helpfulness information in a more comprehensive manner.

Nevertheless, the semantics category contributes largely to the final performance. Throughout this chapter, using UGR alone accounts for 97.96% ± 0.35% of the accuracy compared with the max-performance combination of all features across domains. The exclusive use of SGNS and GV can also yield comparable prediction performance. The empirical results demonstrate that combining many of the selected features, while leading to various performance gains, does not significantly improve helpfulness prediction. Similar to conclusions in [162], UGR plays an important role in helpfulness prediction, whereas the performance modeling combinations of other types of features with/without word-level semantics is not as notable. The findings of this chapter con-

tradicts prior studies largely combining multiple features without solid and sufficient justification. The extensive feature evaluation conducted in this chapter fills the gap of currently arbitrary feature selection process to review helpfulness evaluation.

3.6.1 Practical Guidelines

This section further summarizes general guidelines for feature selection under the three scenarios described above. The summaries can help discover patterns of features or feature combinations for review helpfulness prediction that perform well in general.

The findings and guidelines for review helpfulness prediction using individual features are summarized below:

1. Consider UGR, GV, and SGNS in the semantics category with higher priority since they are the most distinctive for informative reviews. In particular, GV performs better than SGNS in most cases.
2. Features in the sentiment category are less effective in review helpfulness prediction compared with the three semantic counterparts. However, it is worth trying to replace the semantics with LIWC in small datasets.
3. Most features in the structure, readability, and syntax category are of minor predictive power and not suggested to use individually.

The findings and guidelines for review helpfulness prediction using multiple features within each category are summarized below:

1. The max-performance combination of semantic features consistently outperforms those in other categories in helpfulness prediction. Specifically, it is suggested that the combination includes GV as the first feature.
2. Regarding the sentiment (structure) category, it is recommended the max-performance combination base on LIWC (INTERRO+WORD), and subsequently follow an addition of OL+GI (CHAR alone) to the corresponding category since performance gains are reported in most cases.

3. In regard to the readability (syntax) category, it is suggested the max-performance feature combination base on SMOG (ADV), and subsequently follow an addition of one of ARI, CLI, and FKGL (ADJ, NOUN+VERB) to the corresponding category as this generally leads to visible performance gains.

The findings and guidelines for review helpfulness prediction using features from multiple categories are summarized below:

1. Initialize the combination with no more than three (usually one or two) semantic features, starting with GV or UGR, followed by SGNS.
2. Extend the combination with the remaining features in a forward selection manner. It is suggested that features mentioned in Table 3.6 have higher priority than those that are not.
3. Finalize the search by integrating the unused features into the combination using forward selection.

3.6.2 What Makes Review Semantics Stand Out?

The effectiveness of semantic features in helpfulness prediction lies in their direct modeling of review texts. As discussed, the aforementioned categories except semantics used observed statistics, which highly rely on external linguistic resources and can introduce additional noise. For example, the detection of words with misspelling errors is based on the built-in dictionary keeping a static list of manually created words. However, language usage is evolving over times. In a similar vein, the detection of parts-of-speech also suffers from training data biases and/or incomplete predefined rules. Sentiment features can also suffer from unexpected noise due to incomplete and/or incompatible lexicons. Even fine-grained and carefully-designed dictionaries such as LIWC and GI aim at alleviating the potential errors and biases, sentiment lexicons can hardly lead to universal solutions. As pointed out by Yin et al. [325], customers' feelings can be expressed in an implicit yet more complicated manner without using explicit emotional terms. Many existing sentiment lexicons are limited and vulnerable since such context is not taken into consideration. On the contrary, semantic features bypass the external dependency and directly model word meanings from textual information. Therefore,

domain-specific relationships (from shallow to deep) among words and phrases can be captured in an adaptive manner.

From the mathematical perspective, the success of the semantics category can be further explained from two aspects: the encoding dimensionality and encoding methods. UGR, SGNS, and GV encode review content using more dimensions than other features. For many features that only have single dimension, encoding all text information into limited vector space can be challenging. This also explains the reason for category-based sentiment features outperforming those use polarity-based lexicons. On the other hand, both SGNS and GV achieve comparable performance to UGR with far fewer dimensions, showing that the information density of a feature varies from encoding methods. Even when used jointly, features beyond the semantics category are still less representative.

The dominance of review semantics also proves the feasibility of a new helpfulness prediction direction. Instead of laborious feature engineering spending huge efforts in designing counting/peripheral statistics, future researchers can focus on semantic and topical content analysis on review texts and a few closely related observed/derived features. Potential performance gains can be hopefully achieved by modeling sole semantic features from reviews via more advanced techniques, for example, state-of-the-art deep learning algorithms.

3.6.3 Qualitative Investigation

Review semantics, especially unigram information, has been proven to be effective in helpfulness prediction via extensive empirical comparison. In addition to quantitative evaluation, the trained models are further analyzed to understand the effectiveness of review semantics. Recall that linear SVM models learn a hyper-plane that maximally separates the two classes. Once fit with training data, the model can predict the binary helpfulness of reviews by observing which side a data point belongs to. As by-products, the learned model coefficients form a vector orthogonal to the hyper-plane. As such, the direction (i.e., sign) of the coordinates indicates the predicted class and the magnitude (i.e., quantity) reflects feature importance. Such characteristics enable the linear SVM algorithm to qualitatively interpret review semantics.

Table 3.8 demonstrates the 20 most representative words for both helpful and un-

helpful reviews, along with their feature importance. The importance of a word indicates the absolute value of the corresponding vector coordinate, in other words, largest distance to the hyper-plane. An shown in the table, many emotional terms tend be helpfulness related. Among the prominent words, several positive terms (e.g., “outstanding”, “relaxation”, “pleasantly”, “perfectly”, “staggering”, “refreshing”, and “deliciously”) are considered as helpful, whereas negative ones (e.g., “grunts”, “yawn”, “irritating”, “mediocre”, “whining”, “disaster”, “disappointment”, and, “gruesome”) are important for detecting unhelpful reviews. Such phenomenon confirms that emotions embedded in user opinions can empathize readers during helpfulness perception, justifying the large body of existing studies leveraging sentiment analysis techniques for helpfulness prediction. Although sentiment composition show strong indication, the polarity is not necessarily proportional to the perceived helpfulness. For example, positive expressions such as “impressionable” and “acceptable” are found in the unhelpful class; likewise, negative expressions can exist in helpful class, including “degradation”, “refusing”, and “fuming”. This may result from negation, subjunctive mood, or comparison. Depending on the domain context, similar terms (e.g., “paranoia” in D4 helpful and “paranoid” in D1 unhelpful) can occur in different classes.

Another interesting observation is that the term “underrated ” (D4–D5) strengthens review helpfulness, whereas “overrated” and “overblown” (D2–D5) are emphasized in the unhelpful reviews. Similar terms and patterns can be hopefully discovered when increasing the top number of words for both classes. In addition, several potential proper nouns are found influencing the review helpfulness. In D1, the term “lg” (a possible indication for the brand LG) is found in the unhelpful class. In D4, “j-lo” (a possible indication for the celebrity Jennifer Lopez) is found in the helpful class. Other possible abbreviations for companies, organizations, and products are also found, including “pbs” in D1, “teac” in D2, and “v-moda” and “kinect” in D3. Furthermore, informal expressions tend to harm review helpfulness, as the terms “wtf” and “ain’t” in D1, “me2” in D2, and “cuz” in D3 are all found to have high feature importance. Apart from the aforementioned observations, some words seem less reasonable, such as the term “meaning” in D1 and “16g” in D3, which requires further investigation. This qualitative investigation reveals transparent, straightforward, and useful insights into what words are essentially being used by the models to make helpful and unhelpful decisions.

Table 3.8: Prominent words and their importance in the helpful and unhelpful class.

	Helpful	Unhelpful
D1	concern (2.11), bingo (2.08), displays (1.97), outstanding (1.78), switched (1.71), answered (1.71), recipe (1.68), stored (1.65), dragging (1.61), lesson (1.59), winner (1.58), forth (1.57), notch (1.57), relaxation (1.56), flash (1.56), shown (1.56), pbs (1.56), tasks (1.56), unlike (1.55), grew (1.55)	cow (2.17), superman (2.03), paranoid (2.00), ain't (1.87), pile (1.80), offset (1.79), year-old (1.75), payed (1.73), soundtrack (1.70), violence (1.70), priced (1.69), universe (1.66), faotd (1.62), sentence (1.58), acceptable (1.56), whining (1.55), mature (1.52), meaning (1.47), wtf (1.46), lg (1.45)
D2	holes (2.28), expands (2.15), pleasantly (2.07), kindle (1.92), steal (1.92), rack (1.92), helps (1.90), includes (1.86), normally (1.83), assignments (1.82), drawbacks (1.82), episodes (1.79), snipe (1.79), nervous (1.77), surprises (1.77), riven (1.76), tasks (1.76), towers (1.75), entirely (1.75), 2k1 (1.73)	me2 (2.16), grunts (2.00), wears (1.87), selecting (1.76), dev (1.74), yawn (1.72), supposed (1.68), dow (1.62), sucks (1.62), unimpressive (1.61), game-play (1.60), ranges (1.59), atleast (1.59), apps (1.58), overrated (1.56), juice (1.56), engage (1.55), extreme (1.51), defines (1.50), gruesome (1.50)
D3	pleasantly (2.71), perfectly (2.57), hitch (2.45), bonus (2.36), allows (2.03), worried (1.98), iron (1.96), cake (1.93), 150-500 (1.92), refusing (1.90), accurate (1.89), smudges (1.82), pleased (1.80), beat (1.79), magnets (1.79), 16g (1.76), degradation (1.76), flip (1.76), adapter (1.76), teac (1.76)	readynas (1.95), v-moda (1.94), supporting (1.92), tegra (1.90), downgrade (1.84), vine (1.83), cuz (1.81), returned (1.81), returning (1.80), where's (1.78), strikes (1.74), pre (1.74), x20 (1.72), overrated (1.72), disaster (1.71), ridiculous (1.71), judge (1.70), kinect (1.67), awkward (1.67), rave (1.67)
D4	underrated (2.96), must-have (2.40), khia (2.35), molded (2.22), poke (2.17), changer (2.15), rebirth (2.13), realities (2.10), cherone (2.09), frequencies (2.06), heartfelt (2.06), mesh (2.05), addictive (2.03), j-lo (2.03), staggering (2.02), paranoia (1.99), highly (1.98), nazareth (1.98), wandered (1.95), refreshing (1.94)	overrated (2.90), obama (2.67), irritating (2.41), forgettable (2.33), excruciating (2.18), metamorpho (2.08), disappointment (2.02), mockery (1.98), mediocre (1.98), xanadu (1.97), bore (1.95), gruesome (1.95), playlists (1.93), amounts (1.91), prerogative (1.91), over-rated (1.89), fabolous (1.89), soaked (1.88), prozac (1.88), overblown (1.85)
D5	kgharris (3.34), underrated (2.56), saddle (2.50), under-rated (2.49), gem (2.49), true-to-life (2.45), gunner (2.29), pittsburgh (2.29), deliciously (2.29), seige (2.28), refreshing (2.27), rounding (2.26), elections (2.20), wringing (2.16), justly (2.12), delightful (2.09), heartily (2.07), priceless (2.07), rolf (2.07), kirk's (2.05)	prometheus (2.77), overrated (2.65), machete (2.38), undermine (2.32), cavill (2.24), jiggle (2.24), undermined (2.21), marlena (2.20), impressionable (2.20), boring (2.08), mediocrity (2.07), northup (2.07), massie (2.02), endgame (2.01), zod (1.98), wikipedia (1.96), daldry (1.95), t-shirts (1.93), interminable (1.92), sadistically (1.89)
D6	examines (2.51), rut (2.36), lifesaver (2.22), dore (2.16), kildar (2.16), gasped (2.13), mainstay (2.11), curly (2.10), wakeup (2.10), laviolette (2.10), fib (2.07), matrices (2.07), editting (2.07), mini-biographies (2.04), converging (2.02), steven's (2.02), shipyard (2.01), martini (2.01), riot (2.00), fuming (1.99)	klausner (3.77), rautu (3.28), knoxville (3.10), lathan (2.81), falgannon (2.79), macgillivray (2.73), creamer (2.70), chalice (2.63), unbound (2.57), maran (2.36), conspiratorial (2.36), shauna (2.36), garbled (2.35), incidences (2.34), sic (2.33), retinue (2.31), dianetics (2.29), uglier (2.25), vishous (2.22), affliction (2.20)

3.7 Summary

Online product reviews have become an essential source of knowledge for most customers when making e-purchase decisions. In the deluge of data, to identify and recommend the informative reviews, rather than those of random quality is an important task. This chapter presents a computational framework conduct for feature-based helpfulness prediction. Feature-based methods have long been the paradigm of helpfulness prediction due to relatively simple implementation and effective interpretability. In the chapter, the 30 most frequent content-based features from five categories have been identified, and their extensive evaluation is conducted on six top domains of the largest publicly available Amazon 5-core dataset. The individual features, feature combinations within each category, and all feature combinations that lead to max-performance performance have been studied. As stated by Charrada [41], the usefulness of a review is likely to depend on numerous factors that are difficult to isolate and study. The empirical results set comparable and reproducible baselines for review helpfulness prediction, and more importantly, highlight the feature combination patterns that lead to general good prediction performance, regardless of application domain and/or source platform.

Several significant findings and guidelines in feature selection are worth highlighting. Among many features, unigram TF-IDF and the two more recent types of pre-trained word embeddings yield strong predictive power across all domains, demonstrating the effectiveness of encoding semantics for helpfulness prediction. The LIWC dictionary achieves the closest performance to the three semantic features with far fewer feature dimensions, showing the feasibility of helpfulness prediction with fine-grained categorical sentiments. Another important finding is that appropriately increasing the number of features can increase the performance of helpfulness prediction in almost all cases, regardless of feature categories and feature selection scenarios. In particular, combining features from multiple categories effectively improves the performance over individual features, or features from one single category. To summarize, a good rule of thumb for feature selection is to initialize the search with semantic features, followed by features mentioned in Table 3.6, and finally the remaining content-based features. The findings and guidelines of this work can facilitate feature selection in review helpfulness prediction. The exploration of potential factors behind the helpfulness evaluation process will deepen the insights obtained and contribute toward improved prediction system development.

The following directions will be addressed in the future. (1) Selected context-based features and less popular content-based features that are currently excluded will be taken into account to validate their predictive power. Especially, the social connection among reviewers and reviewer characteristics (e.g., reviewer age, the number of history reviews posted by a reviewer) will be emphasized. (2) The potential extension to other domains in the 5-core Amazon dataset and other platforms such as Yelp and TripAdvisor will be included following the holistic view on helpfulness prediction task. Also, the current findings, although generalizable, are based on large datasets. In practice, collecting massive training data can be an expensive task. Since the performance of semantic features may be affected by data size, future work needs to investigate if the findings still hold for small datasets. (3) In addition to the current forward selection search process, more options for computation-friendly feature selection strategies will be explored to search max-performance feature combinations. More statistics (e.g., variance inflation factor [261]) will also be adopted to measure the improvement of feature combinations compared with the individual features. (4) To cope with the minor multi-collinearity issue potentially occurred in the current methodology, additional dimensionality reduction strategies will be applied directly to multi-dimensional features (especially semantic ones) and combinations of one-dimensional features to explore more accurate representations for online reviews. Ngo-Ye et al. [220, 221] report a strong empirical indication that only a selected subset of review words are important in representing review helpfulness. As discussed in [231], many LIWC dimensions are part of a hierarchy, and thus doomed to feature redundancy. For example, "Swear words" is a subcategory of "Informal language". Given the complete use of LIWC dimensions already yield promising results, it is exciting to extract dimension subsets that suffer less from multi-collinearity for helpfulness prediction. (5) The moderating factors will be explored, such as the product type and sequential bias. As stated by Ocampo et al. [224], it is perfectly sensible to expect the helpful reviews of different product types to be different. Given the context of a review, Sipos et al. [280] found that the helpfulness votes are often the consequence of its nearest neighbours. (6) More robust and sophisticated machine learning models will be employed to select representative features for helpfulness prediction. For example, recent explainable deep learning techniques can be employed to model semantic features from review content to free helpfulness prediction studies from heavy feature engineering.

CHAPTER 4

AN INTERACTIVE NETWORK FOR END-TO-END REVIEW HELPFULNESS MODELING

Automatic review helpfulness prediction aims to prioritize online reviews by quality. Existing methods largely combine review texts and star ratings for helpfulness prediction. However, star ratings are used in a way that either has little representation capacity or limited interaction with review texts. As a result, rating information has yet to be fully exploited during the combination. This chapter aims to overcome the two drawbacks. A deep interactive architecture is proposed to learn the Text-Rating Interaction (TRI) for helpfulness modeling. TRI enlarges the representation capacity of star ratings while enhancing the influence of rating information on review texts. TRI is evaluated on six real-world domains of the Amazon 5-core dataset. Extensive experiments demonstrate that TRI can better predict review helpfulness and beat the state of the arts. Ablation studies and qualitative analysis are provided to further understand model behaviors and the learned parameters.

4.1 Introduction

Online reviews play an important role in the e-commerce ecosystem. Currently, online buyers highly rely on collective wisdom to make informed purchase decisions. A recent survey [217] shows that over 8 of 10 customers read reviews for online retailers. The reviews also help manufactures collect user feedback and improve products. Nevertheless, the quality of user-generated reviews is uneven [172], susceptible to customers' background, tolerance of product deficiencies, moods at the time of writing, to name a few. As the number of reviews grows, locating useful information becomes increasingly challenging. Many e-commerce platforms gather user voting on review helpfulness for quality assessment. Still, the voting data is scarce in practice and even missing in less popular products.

Helpfulness prediction aims to identify and recommend high-quality reviews to customers in an automatic manner. Previous literature [224, 117, 41] largely employ review texts and star ratings for the task. The rationale lies in their ubiquitousness in contemporary online shopping platforms and their importance to review helpfulness modeling. Review texts qualitatively describe reviewers' opinions toward product properties. The textual content contains rich information [96], which is an ideal source [75] for learning

★★★★★
Finally, I bought it! This is the best gaming device I could ever dream about. The graphic card is top-notch. Although I came to the store a bit late, the long queue is worth waiting for.

(a) Positive comment with positive rating

★★★★★
Finally, I bought it! This is the best gaming device I could ever dream about. The graphic card is top-notch. Although I came to the store a bit late, the long queue is worth waiting for.

(b) Positive comment with negative rating

Figure 4.1: Consistency between review texts and star ratings can affect helpfulness perception.

helpfulness information. On the other hand, star ratings [199] provide a more straightforward form to quantify reviewers' opinions. The valence (positive or negative) [326] and extremity [275, 230, 87] of ratings are shown to have considerable impact on review helpfulness.

More importantly, the (in)consistency [316, 246, 294] between review texts and star ratings can also affect a consumer's helpfulness perception. The text of a review and its accompanying star rating can be thought of as the qualitative and quantitative aspects [345] of the same user experience. Normally, customers expect the overall opinion of review content to be aligned with the rating [124] during perusal. In practice, however, a review's rating does not necessarily reflect what is mentioned in the content [325] due to the subjectivity of ratings [270, 151]. As a toy example, Figure 4.1 shows two reviews with the same comments but different ratings. In review (b), the mismatching opinions may be considered careless, over-subjective, or being ironic. Such inconsistency is likely to cause confusion and diminish the trustworthiness and thus helpfulness of a review.

Existing methods combine review texts and star ratings for helpfulness modeling to imitate customers measuring the (in)consistency. However, star ratings are used in a way that either has little representation capacity or limited interaction with review texts. In most studies [276, 261, 97], review texts are represented in a high-dimensional feature space, whereas star ratings are used directly. The scalar representation limits the capacity of rating information as well as its influence on review texts. To enlarge encoding capacity, [245] treats star ratings as the final word of its text. The combination is done by learning star embeddings as part of review text encoding, using Convolution Neural Networks (CNNs) as encoders. CNNs operationalize sliding windows on a text to learn features from consecutive words. Under this setting, a star rating only locally

interacts with the last few words of a text and thus has limited interaction. Also, rating information may lose during text encoding due to the max pooling nature in CNNs. As a result, the existing methods are far from fully utilizing rating information.

This work further utilizes rating information for helpfulness modeling. An end-to-end architecture is proposed to learn Text-Rating Interaction (TRI). To enable equivalent representation capacity, TRI maps review texts and star ratings into feature vectors of the same dimensionality. To enlarge the text-rating interaction during combination, text and rating embeddings are first separately learned and then combined. Different from [245] that learns rating vectors as part of content encoding, the encoding of star ratings is decoupled from that of review texts. As a result, rating information can interact with all words in a review text. The decoupling also helps the rating information remain intact and maintain its global influence on review content. The (in)consistency between review texts and star ratings is then captured via the element-wise interaction between content and rating vectors. During the interaction, TRI further adopts gating mechanisms to adaptively learn the amount of rating information needed by review content.

To the best of our knowledge, TRI is the first work that copes with both the representation capacity of rating information and its interaction with review texts for helpfulness modeling. The introduced adaptive rating learning mechanism also allows for more flexibility in leveraging star ratings. TRI is evaluated on six real-world domains of online product reviews. Extensive experiments show that TRI can exploit the text-rating interaction to improve helpfulness prediction and outperforms the state of the arts. Ablation studies and qualitative analysis of the learned model parameters further demonstrate the effectiveness of TRI.

The remainder of the chapter is organized as follows. Section 4.2 surveys related work. Section 4.3 gives the problem statement of interactive helpfulness modeling. Section 4.4 presents TRI and its learning components. Section 4.5 describes experiment settings used to evaluate TRI against a series of state-of-the-art methods. Section 4.6 demonstrates the effectiveness of TRI, conducts detailed ablation studies of the TRI components, and discusses the behavior of TRI via a series of qualitative analysis tasks. Finally, Section 4.7 concludes the chapter.

4.2 Related Work

The topic of automatic helpfulness prediction can be mainly categorized into traditional feature engineering methods and state-of-the-art deep learning approaches. The former has been widely adopted for the past decade, whereas the latter have been recent shown feasible and effective. Overall, review texts have been used as the main source for helpfulness prediction due to the rich information. Combining review texts and star ratings for helpfulness modeling is also gaining increasing attention. This section introduces existing methods using sole review content (Section 4.2.1) and the conjunction of review content and star ratings (Section 4.2.2), respectively.

4.2.1 Content-based Helpfulness Prediction

Various models [322, 192, 291, 182, 97, 141] have been proposed to identify helpful reviews. The mainstream solution is to extract hand-crafted features from r;eview texts, review metadata, and social networks of reviewers, and apply machine learning algorithms to the feature space. Despite these methods have shown to be effective in predicting review helpfulness, the preparation of such features is product- and domain-dependent, which requires tremendous time, prior knowledge, and human effort. Moreover, hand-crafted features are often secondary observed statistics derived from the raw data of reviews. Such features to some extent suffer from multi-collinearity issue [231] (i.e., feature redundancy) and introduce unexpected noise into the prediction process.

The emergence of deep learning [161, 189, 233, 187] using neural networks have achieved some success in modeling helpfulness and is bringing new paradigms into automatic helpfulness prediction. With the help of neural architectures, continuous representations used for model training can be learned automatically, bypassing the procedure of laborious feature engineering [224] used in traditional helpfulness prediction methods. In particular, recent studies based on Convolution Neural Networks (CNNs) [142, 143] have shown the feasibility and effectiveness in learning semantic information from online reviews. Saumya et al. [267] replace the original CNN framework with two convolution layers for helpfulness prediction. The first convolution layer converts a stack of embeddings of word in a review into a document-level representation. The second convolution layer further encodes the review for helpfulness prediction. Experimental results on reviews collected from two shopping platforms show that the presented

model has better capability in learning complex semantics.

In [44], the authors consider a cross-domain task that seeks to predict helpfulness of a domain with limited training data, using knowledge learned from another domain with sufficient reviews. The authors first enrich word embeddings in the CNN framework with character embeddings. Subword information has shown to be effective in alleviating the out-of-vocabulary problem in many applications, especially when training data is insufficient. Three separate CNN layers are built on top of the document embeddings to perform knowledge transferring. In particular, one layer summarizes common knowledge that is shared across domains and the remaining two layers learn domain-specific knowledge. To ensure each CNN layer learns the corresponding knowledge, adversarial loss is defined on the domain-independent layer, whereas domain discrimination losses and negative cross-entropy losses are added to the rest.

Chen et al. [43] extends [44] to study multi-domain helpfulness prediction. Inspired by observations that helpful reviews tend to mention certain product features, the authors incorporate aspect distribution of reviews [321] into word embeddings to learn review representations. Based on the assumption that words in a review may contribute diversely to its helpfulness, the authors propose Embedding-gated CNN (EG-CNN) to identify important/unimportant words in reviews. Specifically, the enhanced word embeddings are fed into a gating layer (a feed forward layer with element-wise sigmoid activation) to learn multi-granularity text features before convolution. Similar to [44], a shared and domain-specific layer is added to model domain commonalities and differences respectively. During model training, the heterogeneous relationship among domains is captured via a domain correlation matrix.

Differ from traditional feature-based helpfulness prediction, the deep learning counterparts tend to pay high attention to learning latent features from sole review texts. The rationale lies in the ubiquitous existence of review texts as a part of the user review generation process in contemporary online shopping platforms. Review content describes reviewer opinions toward product properties [322, 321] where the majority of helpful information is located. Furthermore, review texts containing rich and primary information [96] of reviewers' opinions. [75] conducts extensive experiments to evaluate the effect of frequently used features on helpfulness prediction. Empirical results discover that semantic features are the most useful features for the task. As pointed out by the authors, review semantics allows for more sophisticated modeling of word meanings that other secondary features are incapable of. Therefore, textual content is the most

preferred source for modeling review helpfulness.

In addition to review texts, star ratings [199] is frequently involved as part of the the user review generation process. Review star ratings are chosen due to its simplicity, conciseness, and large existence. The accompanying star rating of a review provides another more straightforward measuring form to summarize reviewers' opinions. Currently, almost all contemporary online shopping platforms adopt five point Likert scale to measure customers' attitudes towards a targeted item. Starting from one star (most unsatisfied) to five stars (most satisfied), the rating suggests increasing satisfaction levels of customers. The raw rating values can be interpreted in multiple methods. Reviews accompanied with one or two stars are usually considered as negative and four or five stars positive; three-star reviews are considered as either neutral or negative. According to [326], the phenomenon termed positive–negative asymmetry plays an important role in human helpfulness evaluation. In terms of online reviews, the star rating of a review (whether positive or negative) will influence its perceived. The extremity of star ratings [275] also affects customers perceiving the helpfulness of online reviews. According to [124], customers expect reviews to be aligned with the overall opinion of a product during perusal. Extreme reviews of which star ratings largely deviate from the average rating usually attract more attention and thus are likely to be perceived as helpful. [230, 87] confirms the U-shaped relationship between review ratings and helpfulness, indicating that extreme ratings are perceived as more helpful than moderate ones.

This chapter proposes a helpfulness prediction framework based upon CNNs [142]. As such, the proposed framework can obtain helpfulness related features in an automatic manner in favor of feature engineering. Following previous work, TRI develops a CNN-based architecture to learn features from review texts. Similar to [43], gating mechanisms are used during content representation learning. Differently, the gates in TRI are not only used to identify word importance but also to combine word embeddings and the convoluted features for multi-granularity content representations.

4.2.2 Interaction between Review Content and Star Ratings

The past decade has seen a large body of studies [224, 117, 41] relying on both review texts and star ratings for helpfulness prediction. In most of the feature engineering approaches [110, 124, 45, 50], rating information is used in conjunction with review texts

by concatenating learned content representations and raw rating values. Several studies [325, 192, 189] consider star ratings as a moderating factor to interact with review texts. In addition to linear rating information, the quadratic term [148, 212, 15] of star ratings is used to validate the influence of rating extremity on the perceived helpfulness.

Although both features are standard measurements of review valence, as Yin et al. [325] state, a review's rating information does not necessarily reflect the customers' opinions mentioned in its content. Since star ratings are subjective reflecting on self needs [270, 151], the valence (positive or negative) of review content can differ from that of star ratings. Jointly using review content and star ratings helps understand customers' opinions in a more comprehensive manner. In practice, the content-rating combination imitates customers' helpfulness voting process. During the process, the qualitative text valence of a review is compared against quantitative rating valence on a product aspect level. As a result, the (in)consistency between review content and star ratings [246, 294] affects a consumer's attitude/trustworthiness towards a review, which is essential to helpfulness prediction.

The essence of jointly using review content and star ratings is their mutual interaction. Currently, three types of approaches are mainly used for fusing the two features. Simple concatenation of both representations is the most used method for feature fusion before feeding the features into prediction models. Several studies interact review content with star ratings to capture more sophisticated relationships between one and the other. Concretely, star ratings are considered as a moderating factor to validate if review helpfulness predicted by a set of content-based features performs differently across rating levels. The rating levels can be either the original five points or further categorized into valence groups by the number of stars. On top of linear rating information, the quadratic term [148, 212, 15] of star ratings is used to capture the effect of extreme reviews on the perceived helpfulness.

More recently, deep learning techniques are used to model the content-rating interaction. Qu et al. [245] propose two CNN variants to combine review content and rating information for helpfulness prediction. The first combination method (CM1) follows traditional machine learning and regards star ratings as an explicit feature. In this case, the extra dimension of raw star ratings is attached to the learned content representations. The second combination method (CM2) treats each star rating as new vocabulary and the last word of a review. The intuition originates in humans evaluating review texts and star ratings at the same time. As such, star ratings are converted into vectors of the

same dimensionality of word embeddings. The word embeddings and star embeddings are then concatenated for learning content embeddings. Experiments on electronic and book-related Amazon reviews reveal that CM2 consistently outperforms CM1.

Instead of combining star ratings into the learned features, Fan et al. [85] formulate review helpfulness prediction as a multi-task neural learning (MTNL) problem. Specifically, a CNN framework is first employed to learn continuous features from review content. Similar to [44], subword information is first adopted to enhance the representation of review content. After convolution operations, each kernel transforms the word embedding matrix into a vector encoding convoluted values. The original CNN framework computes document embeddings by applying max over-time pooling to the convoluted vectors. The authors instead computes the weighted average of the vectors using attention mechanisms. Finally, the learned representation of a review is used to perform two prediction tasks simultaneously: the prediction of review helpfulness which is treated as a classification problem, and that of the accompanying review star rating which is treated as a regression problem. In this work, instead of treating review star ratings as input data, the learned content representations are used as shared features to predict both review helpfulness and raw star ratings.

In [86], the authors consider that the helpfulness of a review should be evaluated within the context of product characteristics. The assumption lies in that a review tends to be more helpful if it contains information mentioned in the targeted product title. For this purpose, a RNN-based framework is proposed to learn continuous representations for product-aware review helpfulness prediction (PRH-Net). First, two sets of bidirectional LSTMs are used to learn separate document representations for review content and the product title. To establish awareness, the product title is matched against reviews on a word level via attention mechanisms. The attention weights show the closeness of review content reflecting on product characteristics, and are used to reinforce review representations. As the final step, the product-aware review representations are fed into the penultimate layer for helpfulness prediction. The training objectives follow [85], which predicts review helpfulness and star ratings simultaneously.

Figure 4.2 depicts the three main methods utilizing review texts and star ratings. In (a), star ratings and learned content representations are concatenated, which cannot capture the mutual interaction between the two features. Even using ratings as a moderator, the weak interaction is constrained by the scalar representation of star ratings. In (b), star ratings are converted into rating embeddings to enlarger encoding space. Still, rat-

ing information has limited interaction with review texts and may lose [135] during the content encoding phase. For example, CM2 interacts ratings with texts through convolution and max pooling. In two extreme cases [45], rating information can dominate the whole representation or do not influence at all. Conversely, star ratings in (c) are used as one of the outputs to be predicted. Such methodology is arguably counter-intuitive because it assumes customers are unaware of rating information when deciding review helpfulness.

To summarize, although combining review texts and star ratings has shown promise in predicting helpfulness, the existing methods either have limited representation capacity of rating information or fail to appropriately establish the text-rating interaction. As a result, star ratings are constrained from providing direct information to content representations. The potential of interacting review texts with star ratings has yet to be fully utilized for helpfulness modeling.

Inspired by [345, 245, 78], TRI embeds star ratings to enlarge the encoding space for rating information. Different from [245], TRI decouples the representation learning of review texts from that of star ratings to avoid possible loss of rating information. To the best of our knowledge, TRI is the first work that takes into account both the encoding and interactive capability of rating information for helpfulness modeling.

4.3 Problem Definition

In this study, helpfulness prediction is formulated as a binary text classification problem. Most existing studies approach the task either by classification or regression. This study adopts the former due to its intuitive and straightforward output (either helpful or unhelpful) to customers.

Let D be a collection of raw online reviews. Each review $d = (r, s, y) \in D$ is a tuple of its text content s, the accompanying star rating r, and the helpfulness label $y \in \{0, 1\}$. The label $y = 0$ indicates an unhelpful review and $y = 1$ helpful. The goal of helpfulness prediction is to learn a classification model F parameterized by θ:

$$F(s, r; \theta) \longrightarrow \hat{y}. \tag{4.1}$$

The model takes as inputs review texts and star ratings, and learns helpfulness information from their interaction. For each review, the model then produces a helpfulness label

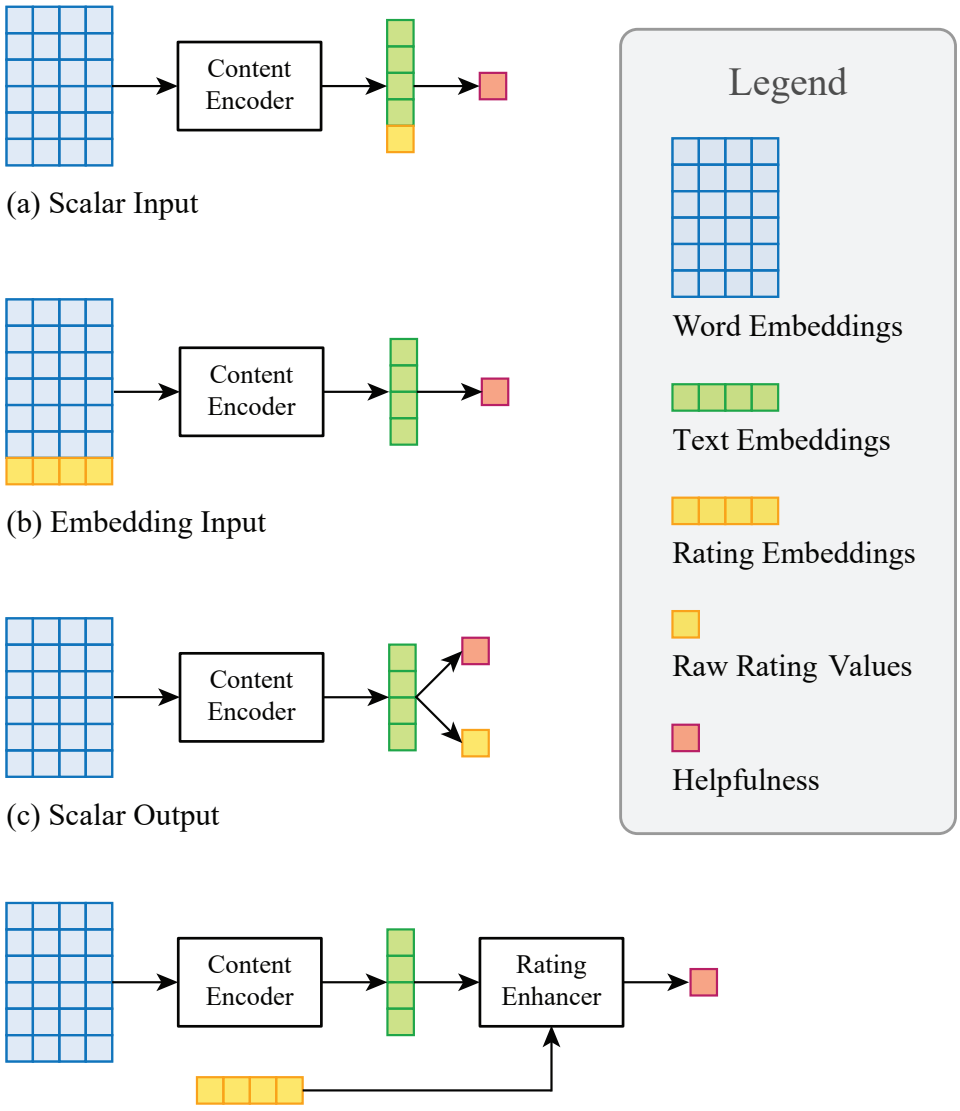

Figure 4.2: Existing approaches combining review content and star ratings.

$\hat{y}$ that approximates the actual helpfulness y of the review.

4.4 Text–Rating Interaction Networks

Figure 4.3 illustrates the TRI architecture. Given a review $d = (r, s, y)$, TRI starts with transforming the text s into an embedding matrix $\mathbf{X}$ and star rating r into an embedding $\mathbf{e}'_r$. Two TRI components are introduced: the content encoder learns content represen-

tations $\mathbf{h}$ from $\mathbf{X}$, whereas the rating enhancer learns adaptive rating representations $\mathbf{r}'$ from $\mathbf{e}'_r$. The two representations are then interacted to jointly learn the rating-enhanced document embedding $\mathbf{h}'$ for helpfulness prediction. The following subsections will give more details of the two learning components.

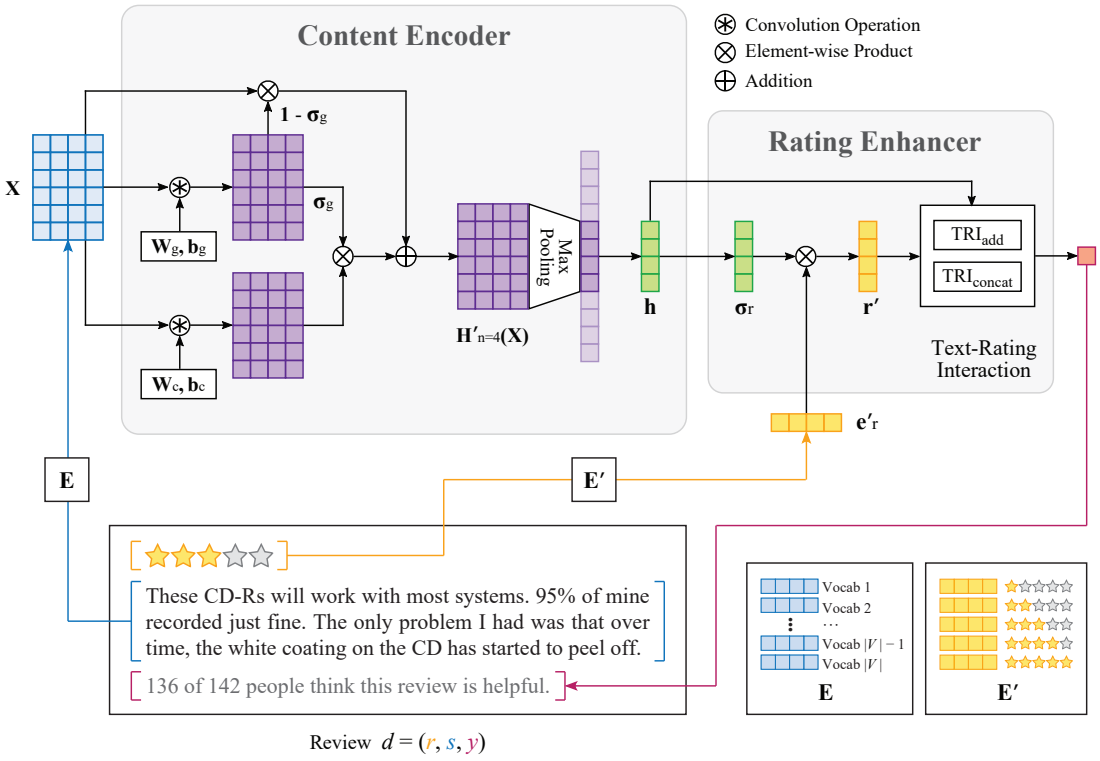

Figure 4.3: The TRI architecture.

4.4.1 Content Encoder

TRI first learns content representations from review texts. Let a review text $s = (x_1, x_2, \ldots, x_N)$ be a sequence of N tokenized words. The content encoder first constructs the vocabulary V by indexing all unique words in D. An embedding lookup table $\mathbf{E} \in \mathbb{R}^{|V| \times d}$ is employed to associate each word x in the vocabulary with a d-dimensional vector $\mathbf{e}_x = \mathbf{E}^\top \mathbf{x}$, where $\mathbf{x} \in \mathbb{R}^{|V|}$ is the one-hot encoding of the word x. Therefore, a text s can be represented by an embedding matrix $\mathbf{X} \in \mathbb{R}^{N \times d}$ by simply stacking the embedding of the constituent words:

$$\mathbf{X} = [\mathbf{e}_{x_1}, \mathbf{e}_{x_2}, \ldots, \mathbf{e}_{x_N}]. \tag{4.2}$$

The embedding matrix $\mathbf{X}$ of a text is used for hidden semantics extraction. Previous work has predominantly used Recurrent Neural Networks (RNNs) [170] for text encoding due to the sequential nature. In RNNs, a memory of occurring information in a sequence is maintained. Training such memory is computationally inefficient as it cannot be parallelized over sequential tokens. One alternative is using Gated Linear Units (GLUs) [69], which allows for parallelization while maintaining a large range of memory. As such, GLUs can be thought of as a faster implementation that approximates the behavior of RNNs.

The TRI content encoder is developed upon GLUs for efficient training. Specifically, GLUs apply two sets of CNN kernels $\mathbf{W}_c$ and $\mathbf{W}_g$ of identical shape to learn separate convoluted matrices from $\mathbf{X}$. The values of one matrix are normalized into $[0, 1]$ and then multiplied by that of the other to obtain the feature maps $\mathbf{H}(\mathbf{X}) \in \mathbb{R}^{N \times d}$:

$$\mathbf{H}(\mathbf{X}) = (\mathbf{X} * \mathbf{W}_c + \mathbf{b}_c) \otimes \boldsymbol{\sigma}_g, \tag{4.3}$$

$$\boldsymbol{\sigma}_g = \sigma(\mathbf{X} * \mathbf{W}_g + \mathbf{b}_g), \tag{4.4}$$

where σ is the sigmoid function and $\otimes$ the Hadamard product between matrices. The kernels $\{\mathbf{W}_c, \mathbf{W}_g\} \in \mathbb{R}^{nd \times d}$ and biases $\{\mathbf{b}_c, \mathbf{b}_g\} \in \mathbb{R}^d$ are parameters to be estimated. Each of the d kernels slides over $\mathbf{X}$ to compute the convolution on n consecutive word embeddings $\mathbf{e}_{x_{i-n+1}}, \mathbf{e}_{x_{i-n+2}}, \ldots, \mathbf{e}_{x_i}$, where $0 < i < N + n$. The missing embeddings are replaced by zero vectors when $i < 0$ and $i > N$.

The use of GLUs for encoding review texts is advantageous. First, GLUs facilitate the training process by allowing gradients to flow through the encoder layers. During back propagation, the first addend of the gradient $\nabla\mathbf{H}(\mathbf{X}) = \nabla(\mathbf{X} * \mathbf{W}_c + \mathbf{b}_c) \otimes \boldsymbol{\sigma}_g + (\mathbf{X} * \mathbf{W}_c + \mathbf{b}_c) \otimes \boldsymbol{\sigma}_g^{'} \nabla(\mathbf{X} * \mathbf{W}_c + \mathbf{b}_c)$ provides a linear path that maintains the scale of the activated gating units. Such a structure reduces the gradient vanishing problem in neural networks as more layers are stacked. The linear path can also be thought of as a multiplicative skip connection [112] between encoder layers. Secondly, the values of $\boldsymbol{\sigma}_g \in [0, 1]$ enable gating mechanisms on the convoluted features. In this case, each of the encoded (convoluted) word embeddings is bound with a gate indicating different word importance. This resembles the use of gated word embeddings in [43] for multi-granularity text features. Thirdly, $\boldsymbol{\sigma}_g$ are further utilized to merge the word embeddings (low-level information) and feature maps (high-level information):

$$\mathbf{H}^{'}(\mathbf{X}) = \mathbf{H}(\mathbf{X}) + (1 - \boldsymbol{\sigma}_g) \otimes \mathbf{X}. \tag{4.5}$$

Here, the gates $\boldsymbol{\sigma}_g$ estimate the ratio of low- and high-level information required. From the perspective of GRU [51], the combination can also be thought of as determining how much new information $\mathbf{H}(\mathbf{X})$ is used to update the previous memory $\mathbf{X}$ at each time step. Setting the values of $\boldsymbol{\sigma}_g$ to 1 considers only the feature maps. In contrast, $\boldsymbol{\sigma}_g = 0$ indicates the exclusive use of the word embeddings.

In TRI, kernels of patch size $n = \{3, 4, 5\}$ are used simultaneously to learn hidden semantics from n-grams in a review. Column-wise max-overtime pooling [60] is then applied to obtain the most salient features. Finally, the pooled features are concatenated and projected via learnable parameters $\mathbf{W}_h \in \mathbb{R}^{3d \times m}$:

$$\mathbf{h} = [\, \max\{\mathbf{H}^{'}_{n=3}(\mathbf{X})\}, \max\{\mathbf{H}^{'}_{n=4}(\mathbf{X})\}, \max\{\mathbf{H}^{'}_{n=5}(\mathbf{X})\}]\mathbf{W}_h, \tag{4.6}$$

where $[,]$ concatenates the pooled feature vectors. As a result, the continuous vector $\mathbf{h} \in \mathbb{R}^m$ represents a review text.

4.4.2 Rating Enhancer

Subsequently, rating information interacts with the content representation. Without loss of generality, a K-point Likert scale is assumed for rating. The scale ranges from 1 (least satisfied) to $K \in \mathbb{N}^+$ (most satisfied), expressing the level of customers' satisfaction towards an item. Let $R = \{1, 2, \ldots, K\}$ be the collection of all possible star ratings. For instance, Amazon adopts a five-point Likert scale for star rating, and hence a rating that accompanies a review can be one of $R = \{1, 2, 3, 4, 5\}$.

Similar to the embedding process of word vectors, each possible rating $r \in R$ is first converted to its the one-hot encoding $\mathbf{r} \in \mathbb{R}^{|K|}$. The associated m-dimensional vector $\mathbf{e}^{'}_r = \mathbf{E}'^{\top}\mathbf{r} \in \mathbb{R}^m$ of a review is then obtained via another lookup table $\mathbf{E}^{'} \in \mathbb{R}^{K \times m}$. Compared with raw ratings (i.e. scalars), rating embeddings allow for m times larger capacity for encoding rating information. Moreover, the vectorization leads to higher representation robustness since any possible noise resided in a raw rating is distributed into individual dimensions.

The rating embedding $\mathbf{e}^{'}_r$ is set to have the same dimensionality as the content representation $\mathbf{h}$ to perform element alignment. As discussed, review texts and star ratings can be thought of as two measurements of the same user experience. While both measurements take different forms of output (i.e., words versus a scalar), the latent evaluation

criteria that lead to the decisions are highly similar. This work hypothesizes that such criteria are reflected by individual embedding dimensions. Thus, aligning $\mathbf{h}$ and $\mathbf{e}_r^{'}$ forces the network to encode each criterion into the same dimension in both embeddings.

The text-rating interaction is established in two steps. In the first step, the star rating of a review is adjusted according to its text. In reality, a star rating has various influences on customers' helpfulness perception depending on what the review text mentions. Each element of a learned content representation $\mathbf{h}$ thus requires rating information differently from the corresponding dimension in the rating embedding $\mathbf{e}_r^{'}$. To perform such estimation, a fully-connected gating layer is built upon $\mathbf{h}$:

$$\boldsymbol{\sigma}_r = \sigma(\mathbf{W}_r^{\top}\mathbf{h} + \mathbf{b}_r), \tag{4.7}$$

$$\mathbf{r}' = \mathbf{e}_r^{'} \otimes \boldsymbol{\sigma}_r. \tag{4.8}$$

The adaptively learned ratios $\boldsymbol{\sigma}_r$ (parameterized by the weights $\mathbf{W}_r \in \mathbb{R}^{m\times m}$ and biases $\mathbf{b}_r \in \mathbb{R}^m$) are then used to adjust the rating embeddings in an element-wise manner. The adjusted rating embedding imitates a more realistic situation that review texts may have sway over customers' perception of star ratings. Note that setting the ratios $\boldsymbol{\sigma}_r = \mathbf{1}$ uses rating information with no adjustment, whereas $\boldsymbol{\sigma}_r = \mathbf{0}$ ignores star ratings.

In the second step, the content representation $\mathbf{h}$ is combined with the adjusted rating embedding $\mathbf{r}'$. Review texts often contain emotional words expressing user experience. As a result, content representations are encoded with certain forms of internal emotions. Given that the emotions in review texts and star ratings can be expressed differently, the compatibility between the two sources should be taken into consideration. TRI explores two combination methods.

- **Addition**. The first method assumes that the internal emotions in review texts and rating information tend to be more homogeneous. In this case, $\mathbf{r}'$ can be thought of as element-wise residual correction or refinement on the emotional components embedded in $\mathbf{h}$.

$$\mathbf{h}' = \mathbf{h} + \mathbf{r}'. \tag{4.9}$$

- **Concatenation**. The second method assumes less homogeneity between the internal emotions and rating information. In this case, $\mathbf{r}'$ serves as new information by supplying $\mathbf{h}$ with additional dimensions.

$$\mathbf{h}' = [\mathbf{h}, \mathbf{r}']. \tag{4.10}$$

The interactive vector $\mathbf{h}'$ represents a rating-enhanced review text. For simplicity, the two methods are henceforth called TRI_{Add} and $\text{TRI}_{\text{Concat}}$, respectively.

4.4.3 Training Objective

Finally, the rating-enhanced content representation $\mathbf{h}'$ is forwarded into a dropout layer, followed by logistic regression to predict the helpfulness $\hat{y}$ of one review:

$$\hat{y} = \sigma(\mathbf{W}_o^\top \mathbf{h}' + b_o), \tag{4.11}$$

where $\mathbf{W}_o^\top \mathbf{h}'$ is defined as $\mathbf{W}_o^\top \mathbf{h} + \mathbf{W}_o^\top \mathbf{r}'$ in TRI_{Add} and $\mathbf{W}_{o1}^\top \mathbf{h} + \mathbf{W}_{o2}^\top \mathbf{r}'$ in $\text{TRI}_{\text{Concat}}$. From a mathematical perspective, TRI_{Add} is a special case of $\text{TRI}_{\text{Concat}}$ when the two halves of the weight matrix $\mathbf{W}_{o1}$ and $\mathbf{W}_{o2}$ are identical. Given M training samples, TRI is learned via cross-entropy minimization:

$$\mathcal{L} = -\frac{1}{M}[\mathbf{y}^\top \log(\hat{\mathbf{y}}) + (1 - \mathbf{y})^\top \log(1 - \hat{\mathbf{y}})], \tag{4.12}$$

where $\hat{\mathbf{y}}$ are the predicted helpfulness labels and $\mathbf{y}$ actual helpfulness labels.

4.5 Experiment Settings

This section conducts extensive experiments to quantitatively and qualitatively evaluate TRI. Section 4.5.1 gives a brief introduction to the datasets and pre-processing steps used throughout the experiments. In Section 4.5.2, the baselines using both traditional machine learning algorithms and state-of-the-art deep learning methods are described for performance comparison. Section 4.5.3 presents hyperparameters for training TRI and the baseline models.

4.5.1 Datasets

TRI is evaluated on the Amazon 5-core dataset [113], one of the largest datasets that are publicly available for helpfulness prediction tasks. Amazon is the largest Internet retailer, which has accumulated large-scale user-generated reviews. The helpfulness of such reviews is rated by online customers, which makes it an ideal candidate for

review helpfulness prediction task. In fact, Amazon product reviews are predominantly used and analyzed in previous studies. Thus, adopting Amazon reviews allows for fair comparisons with previous studies. Also, the analysis results can hopefully provide practical insights into the context of online business and user-generated content quality evaluation.

The original dataset consists of 24 domains, covering 142.8 million reviews collected between May 1996 and July 2014. In this chapter, six domains having the highest number of reviews are selected for evaluation. The domains include Apps for Android, Video Games, Electronics, CDs and Vinyl, Movies and TV, and Books. For simplicity, the first domain is called D1, the second D2, and so on. The large number of online reviews is to ensure sufficient training data for the data-hungry deep learning architectures. Table 3.2 presents a review sample randomly select from the domain of Video Games, along with the accompanying attributes. As depicted in the table, each review contains a set of attributes, including (1) the ID of the targeted product, (2) the helpfulness information, namely the number of helpful and unhelpful votes given by online customers, (3) the star rating of the review, (4) the published date, week, and time of the review, (5) the ID and name of the reviewer, and (6) the summary headline and review text commenting in detail on the product. This chapter focuses on the textual content and star rating of a review. The former results from the concatenation of review summary and review text.

The vote distributions are presented in Figure 3.2, displaying a similar pattern for each domain that high frequency of reviews have a relatively low number of votes. Subsequently, Figure 4.4 demonstrates the review length (i.e., the number of words) dispersion across domains via box plots. The long tail effect is observed in all domains: the length of most reviews is within a certain range, with a small number of outliers being unusually longer. From an implementation perspective, TRI and other neural baselines only accept review inputs with a fixed length. As will be discussed, this chapter will normalize the online reviews to ensure that the inputs are of identical length. Finally, Figure 4.5 presents the rating distributions across domains. As shown, customers tend to give positive feedback, with five-star (four- and five-star) ratings accounting for over half (70 percent) of the reviews. This phenomenon is identified as positivity bias [196] in accordance with many existing studies [172, 54, 229, 246].

Table 4.1 showcases helpful versus unhelpful review examples across the six domains. It can be seen that even similar opinions/attitudes expressed in review texts can

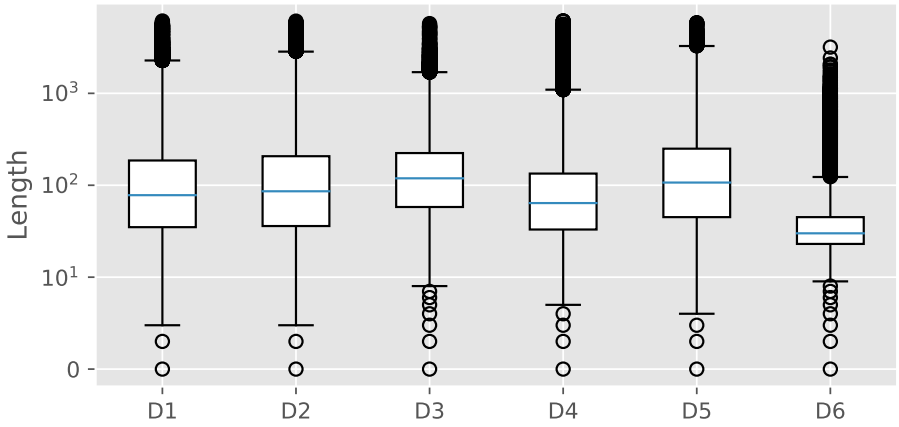

Figure 4.4: Review length distributions.

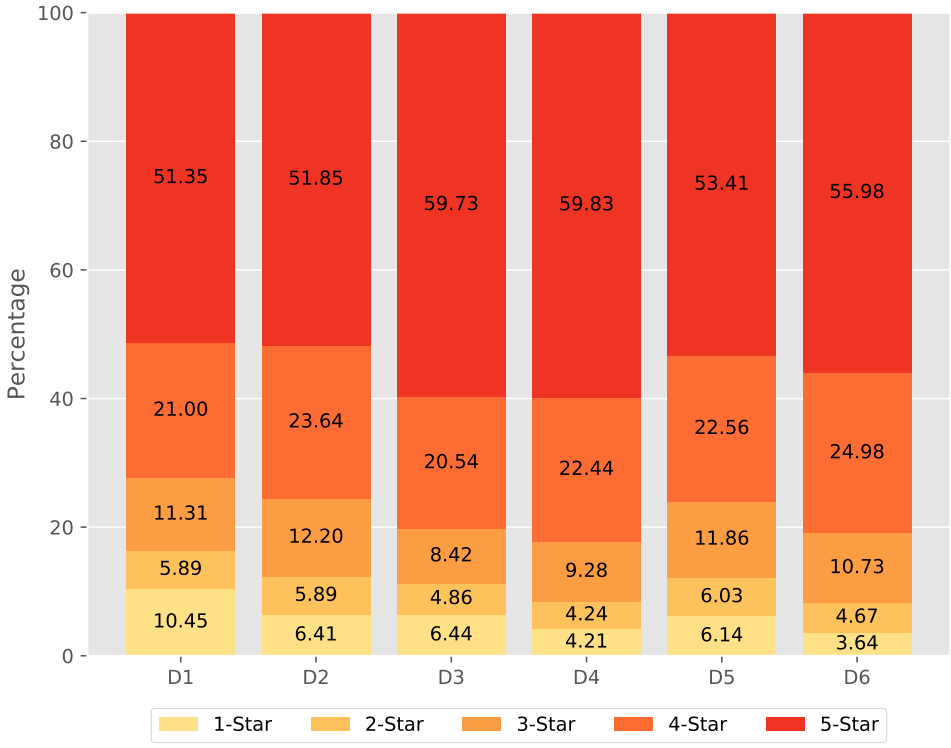

Figure 4.5: Review rating distributions.

receive far different star ratings. This phenomenon commonly occurs in online environment since customers give ratings based on their own standards, which are susceptible to customers' experience, education background, tolerance towards product deficiencies, moods at the time of writing reviews, to name a few. Such inconsistency can affect customers in perceiving the helpfulness of reviews. The difference between review writers' attitudes and readers' prior expectation can further influence the perceived helpfulness. For example, one may think mismatching reviews are rather careless or over-subjective, and thus less trustworthy.

To improve data quality, the following pre-processing steps are applied to the raw online reviews. (1) This chapter focuses on English review helpfulness prediction. For this purpose, blank and non-English reviews are filtered out. (2) Identical and nearly identical reviews [68] are common on Amazon. To avoid training data redundancy, only the ones with the highest number of votes are retained. Following the definition given by [141], two reviews are nearly identical if more than 80 percent of their bigram occurrence is shared. (3) Several reviews only have few votes and the prediction of which may lead to biases that only reflect a small group of people' attitudes towards review helpfulness. To alleviate the effect of words of few mouths [258, 334], reviews with less than 10 votes are skipped. (4) Each of the remaining reviews is lowercased and tokenized in to a sequence of words. Minimum stopword removal is applied by eliminating articles (i.e., a, an, and the) from the reviews. Further stopword removal is not considered since some stopwords can be useful in building review helpfulness. For example, negation expressions such as "not" and "never" are often considered as stopwords, which flip the opinions written by customers. (5) To ensure the training process of neural models does not consume excessive computational resources, only the most frequent 30k terms are kept as vocabulary. In the preliminary experiments on small datasets D1 and D2, using top 30k terms for helpfulness prediction achieves similar performance to using the full vocabulary, which shows the feasibility in training models without less frequent terms. As for review length normalization, the fixed length N for each domain is set to the one whose word count is larger than 90 percent of the reviews.

The pre-processed reviews are then labeled prior to model training. The helpfulness label of the pre-processed reviews is determined in an automatic manner using existing human assessment provided by online users. One standard method is based on the ratio of helpful votes, which is the number of helpful votes divided by the total number of

Table 4.1: Example helpful and unhelpful reviews. Typos and grammatical errors are intentionally preserved.

	Helpful	Unhelpful
D1	421 of 516 people think this review is helpful. ★★★★★ **cute** This is a cute app but after visiting the desert to serve the ice cream, which is only the second area to serve, you have to shell out money to advance because you have to add more flavors to the menu. [...]	3 of 131 people think this review is helpful. ★★★★★ **are you serious?!!?** I love hangman personally. when I saw this app I was like hmmm... and then NINETYNINE DOLLARS ARE YOU KIDDING ME? I don't care how ultimate this is. no app is worth 99$ [...]
D2	153 of 178 people think this review is helpful. ★★★★★ **Careful don't pay over $30.** Available from Amazon.com for $29.99 be wary of first listing. Game is QTE heavy and cinematic. Check the reviews for the standard version. This game is no [...]	8 of 54 people think this review is helpful. ★★★★★ **All Looks and No substance** This game was a huge disappointment for me. After I saw the trailers I was excited to play it but the campaign was short and the first few chapters was fun but afterwards its the same crappy [...]
D3	136 of 142 people think this review is helpful. ★★★★★ **Good, But a Word of Warning...** These CD-Rs will work with most systems. 95% of mine recorded just fine. The only problem I had was that over time, the white coating on the CD has started to peel off. [...]	11 of 64 people think this review is helpful. ★★★★★ **It's a good lens but...** Bought the 300 f/4 thought it would be a good addition since it was light and image quality was good.Got tired of it pretty quick though. F/4 just isn't wide enough for what I wanted to [...]
D4	66 of 81 people think this review is helpful. ★★★★★ **Worst Remasters Ever...** All I want to say is these new Stones remasters are brittle...washed out...compressed...totally distorted and unlistenable. Do not buy these...stay with your Virgin Records versions... [...]	14 of 76 people think this review is helpful. ★★★★★ **BORING BACH!** Pierre Fournier's own interpretation of the unaccompanied cello suites really put me to sleep. Okay, he may be one of the greatest cellists in this generation, but I could care less about this album. [...]
D5	203 of 206 people think this review is helpful. ★★★★★ **DVD is missing 30 minutes** This 1978 British television production is one of the better English-language adapatations of Les Miserables. Unfortunately, the DVD release is missing 30 minutes of footage. [...]	10 of 120 people think this review is helpful. ★★★★★ **AWFUL!** This is soapy garbage. Just how much sex are these doctors having? Probobly more than the doctors on ER and the ones onthe long gone Chicago Hope. This, like HOUSE, M.D., is a medical show [...]
D6	92 of 98 people think this review is helpful. ★★★★★ **A most excellent sourcebook.** Every theologian, occultist, and pious scholar should get this. Virtually every angel, spirit, devil, and lowly demon is named and defined. It also includes a vast list of alternate spellings and comparisons between the [...]	10 of 92 people think this review is helpful. ★★★★★ **People got a lot of nerve** I am pretty amazed by the number of people that say this is what I want to happen in the book or this is how I think this book should proceed!. Oh really!!! Attention all current and future [...]

votes. Subsequently, the continuous ratio is then converted into binary helpfulness labels via a pre-defined threshold. This chapter sets the threshold to 0.6, which is the most commonly used threshold in prior research [97, 152, 189]. For each domain, a review is labeled as helpful if its ratio of helpful votes is equal to or more than 0.6, indicating that at least 60% of users believe the review is helpful. Otherwise, reviews with ratio less than 0.6 are labelled as unhelpful. To avoid the class imbalance problem, which is outside the scope of this chapter, helpful reviews are randomly sampled to have the same number as unhelpful ones and vice versa.

Reviews in each domain are partitioned using a unified scheme. Specifically, stratified random split is adopted: the whole collection of reviews is first shuffled (with a fixed seed), and then 80%, 10%, and 10% of the reviews are randomly selected respectively to build the training set, validation set, and testing set. During the selection, the percentage of samples for each class is preserved. Throughout the chapter, TRI and all baseline models are trained on the training set, tuned on the validation set, and evaluated on the test set serving as unseen data in reality. After dataset partition, review words that are numeric values are replaced by the term <NUM>. For each domain, the term <UNK> is used to alter OOV words (viz. terms that exist in the training set but are missing from validation/test set) in the reviews. Table 3.3 demonstrates the simple descriptive statistics (including OOV rate) of the six domains sorted by data size in ascending order. More descriptive statistics and discussions of the six domains can be found in Appendix 7.

4.5.2 Baseline Methods

TRI is benchmarked against twelve baselines, including seven traditional machine learning methods and five state-of-the-art deep learning architectures. The traditional machine learning models include unigram TFIDF representations, three types of pre-trained word embeddings, two types of categorical sentiment dictionaries, and sole review star ratings. The deep learning methods for helpfulness prediction include the vanilla CNN framework and extended variants using rating information. Note that the PRH-Net model [86] uses extra product information for training, which is unfair to TRI and thus skipped.

- **TFIDF+SVM**: Unigrams have been proved robust and effective in many text min-

ing applications. This baseline trains linear SVM classifiers on TFIDF representations of review unigrams, where terms with document frequency fewer than 1% of the training samples are ignored.

- Recent word embeddings learned from shallow neural networks also show promising performance. Following [75], three types of pre-trained embeddings are used. SVM classifiers are then trained on review representations. The embedding of a review is the average of that of its constituent words, where out-of-vocabulary words are ignored.
  - **SGNS+SVM**: This baseline adopts the 300-dimensional distributed embeddings [201] trained on 100 billion words from Google News.
  - **GV+SVM**: This baseline adopts the 300-dimensional Global Vectors [237] trained on 840 billion words from Common Crawl.
  - **DS+SVM**: This baseline employs the Skip-gram model [201] to train domain-specific word embeddings on each domain of the pre-processed reviews.
- Sentiment analysis also shows strengths in modeling helpfulness prediction. Following [75, 322], two fine-grained sentiment dictionaries are considered. SVM classifiers are then trained on extracted sentiment features.
  - **LIWC+SVM**: The Linguistic Inquiry and Word Count dictionary [236] presets 93 categories for contemporary English, including social and psychological states. The dictionary covers almost $6,400$ words, word stems, and emoticons.
  - **GI+SVM**: General Inquirer [129] attaches syntactic, semantic, and pragmatic information to part-of-speech tagged words. The dictionary contains $11,788$ words assigned to 182 specified categories.
- **RAT+SVM**: This baseline trains linear SVM classifiers on the sole star rating information of reviews.
- **CNN** [142]: The vanilla CNN architecture for sentence classification.
- **EG-CNN** [43]: A variant of the vanilla CNN architecture where character embeddings and word-level embedding gates are used before convolution.
- **CM1** [245]: A variant of the vanilla CNN architecture where raw rating values and content representations are concatenated.

- **CM2** [245]: A variant of the vanilla CNN architecture where rating vectors and word embeddings are concatenated to learn content representations.
- **MTNL** [85]: A variant of the vanilla CNN architecture for multi-task learning, with character and word embeddings as inputs, attention on the convoluted feature maps, and raw rating regressing as the secondary task.

4.5.3 Hyperparameters

The hyperparameters used for training TRI are described as follows. The lookup table $\mathbf{E}$ in neural architectures is initialized with domain-specific word embeddings. Once initialized, $\mathbf{E}$ is kept non-static during training in the CNN baseline and static in other neural architectures, which is determined by the validation set of each domain. The lookup table $\mathbf{E}^{'}$ for mapping raw star ratings is randomly initialized from a uniform distribution in the range $[-0.05, 0.05]$.

Inside TRI, the content representation dimensionality is set to $m = d = 200$. The rating scale of $K = 5$ levels is adopted following Amazon and many contemporary e-commerce platforms. Rectified linear units are used for feature activation. Dropout operations of rate 0.5 are conducted on the penultimate layer to randomly mask half of the layer outputs. The remaining network weights are initialized using the Glorot uniform initializer [100]. Neural weights are updated through stochastic gradient descent over shuffled mini-batches using the mini-batch size of 64 and the Adam [145] update rule. Early stopping occurs when the validation loss has no improvement for 10 epochs.

The other neural baselines are re-implemented following the original hyperparameter setting in the papers except for word vector initialization. The penalty term C in SVM is chosen via a grid search of $\{0.01, 0.1, 1\}$. In cases where raw star ratings are used (either alone or in conjunction with other features), the values in R are normalized into values between 0 and 1 via $\{\frac{1}{K}, \frac{2}{K}, \ldots, \frac{K}{K}\}$. The normalization helps prevent raw rating values from distorting differences in the ranges of values of other features. For $K = 5$, the normalized star ratings are $R = \{0.2, 0.4, 0.6, 0.8, 1\}$. The normalization helps prevent raw rating values from distorting differences in the ranges of values of other features.

For result reproducibility, all randomization processes involved are initialized with the same random seed. For result reliability, all neural models are trained and evaluated

five times on each domain to report the average accuracy. SVM-based models are run once since the results are deterministic.

4.5.4 Implementation

This chapter compiles the experiments using the Python (version 3.6) programming language. The code implementations are run on Ubuntu 16.04.5 Long Term Support as the system environment. The main hardware configuration is as follows: Intel Core i5-9600K CPU @ 3.70GHz $\times$ 6, Samsung SSD 970 EVO, and 32GB RAM. In particular, Nvidia GeForce RTX 2070 is used to accelerate the computation- and bandwidth-hungry neural network training. Parallel computation is enabled via the CUDA (version 9.0.176) toolkit.

In this chapter, text pre-processing and feature extraction are done using NLTK [27]. As for the LIWC baseline, the Linguistic Inquiry and Word Count dictionary (version 2015) [236] is used for feature extraction, whereas the General Inquirer basic spreadsheet from the official website is used to build the GI baseline. The TFIDF modeling and linear SVM classifier [297] is developed using Scikit-learn [235]. The public Python library Keras [53] is used for deep learning model construction. The training of domain-specific word embeddings is fulfilled via the open-source Python topic modeling library Gensim [251].

4.6 Result Analysis and Discussions

This section empirically evaluates the proposed TRI framework from several perspectives. In Section 4.6.1, a series of tasks are conducted to check the sanity of the pre-trained domain-specific word embeddings prior to model training. Section 4.6.2 demonstrates the effectiveness of TRI. Section 4.6.3 performs ablation studies to validate the TRI components. Section 4.6.4 compares the two rating enhancement methods TRI_{add} and $\text{TRI}_{\text{concat}}$, and discusses their performing behaviors. Finally, Section 4.6.5 provides qualitative analysis on the learned model weights (i.e., document embeddings, rating embeddings, and adaptive rating gates), followed by case studies.

4.6.1 Sanity Check

TRI employs domain-specific word embeddings to initialize the CNN-based framework. As shown in many applications [336, 43, 85, 311], the initialization of pre-trained word meanings is of vital importance in training and subsequent tasks. To ensure the quality of the employed embeddings in terms of word similarity, sanity check [176, 142] is conducted. Specifically, for each domain, the ten closest vectors are retrieved for a sample of words to check if the trained embeddings can learn meaningful specifics.

Here, a total of nine seed words are selected for examination, including four general words ("good", "bad", "convenient", "inconvenient" and five domain-related words ("cheap", "expensive", "quality", "price", "discount"). In addition, two popular general purpose counterparts are compared: (1) Distributed representations learned using Skip-Gram with Negative Sampling (SGNS) [201] and (2) Global Vector (GV) [237] trained on the non-zero entries of a global word-word co-occurrence matrix. The comparison is to observe the difference between domain-specific embeddings trained on online product reviews and those trained on more general textual materials.

The detailed validation and notable observations are discussed in Appendix 8. The discussions prove that the trained embeddings for each domain are sane and valid for TRI parameter initialization.

4.6.2 Comparison with Baseline Methods

Table 4.2 reports the prediction accuracy of TRI against the baselines in helpfulness prediction. The bold results indicate models achieving the highest accuracy in each domain. TRI results higher than the baselines are in italics. Two-sided independent t-tests are computed between TRI and the baselines to validate the null hypothesis that TRI_{add} and $\text{TRI}_{\text{concat}}$ significantly outperform the existing state-of-the-art methods.

In brief, TRI outperforms the baselines by approximately 1%–5% in accuracy across domains. Both TFIDF+SVM and RAT+SVM set strong baselines for helpfulness prediction. The three types of pre-trained word embeddings SGNS+SVM, GV+SVM, and DS+SVM achieve comparable performance to TFIDF+SVM, with far fewer dimensions at the price of about 1% loss in accuracy across domains. In particular, DS+SVM produces the highest performance, showing the necessity of using domain-specific word

Table 4.2: Results of TRI against other methods.

Model	D1	D2	D3	D4	D5	D6
TFIDF+SVM	67.68	76.71	75.66	82.52	78.58	75.03
SGNS+SVM	67.58	75.15	73.28	81.02	77.26	73.91
GV+SVM	68.66	74.94	73.34	81.06	77.41	74.04
DS+SVM	68.76	75.54	74.72	81.97	77.92	74.32
LIWC+SVM	66.16	73.94	70.78	76.99	72.25	68.58
GI+SVM	63.76	69.18	67.07	72.07	70.75	67.04
RAT+SVM	70.47	77.45	78.08	85.85	82.13	78.15
CNN	70.38	77.60	77.50	84.04	80.76	77.81
EG-CNN	70.60	78.21	78.63	85.01	81.50	78.38
CM1	71.09	77.82	78.58	84.85	81.37	78.26
CM2	71.00	77.99	79.37	85.39	81.49	78.52
MTNL	67.79	75.60	75.21	82.45	78.42	75.72
TRI_{add}	**72.24*****	**79.00*****	*80.06***	*87.01****	**83.58*****	*80.45****
$\text{TRI}_{\text{concat}}$	*72.04****	*78.37*	**80.22*****	**87.22*****	*83.50****	**80.57*****

* $p < 0.1$.
** $p < 0.05$.
*** $p < 0.01$.

embeddings for neural model initialization. The two sentiment baselines LIWC+SVM and GI+SVM, however, are the worst among all baselines, suggesting that review sentiment alone may be insufficient for helpfulness learning.

The neural architectures except MTNL outweigh traditional ones in learning helpfulness information. CM2 on average achieves the closest performance to TRI. As will be discussed in Section 4.6.3, the effectiveness of CM2 is due to the vectorized encoding of rating information, which can be thought of as an implicit form of text-rating interaction. This again confirms that combining review content and ratings can assist in learning more expressive helpfulness information. Surprisingly, MTNL is worse than CNN and even traditional baselines in certain domains. The mediocre results require further investigation on the influence of review domains, data size, and model hyperparameters on model performance.

The effectiveness of TRI demonstrates the importance of the text-rating interaction. As discussed, review texts expresses the qualitative aspects of user opinions. The same opinion can also be measured quantitatively by the accompanying star rating. Whether the two perspectives are consistent can influence readers in perceiving review helpful-

ness. TRI aims at capturing such consistency during helpfulness modeling, which leads to improvement over the baselines. In the following subsections, the learned interactions will be discussed in further detail.

4.6.3 Ablation Studies

The following four TRI variants are considered to better understand the model behavior. Each variant disables a learning component of TRI to validate the change of model performance. Table 4.3 illustrates the accuracy of the four variants. Overall, TRI outperforms any of its variants, showing the necessity of the proposed TRI learning components to achieve the performance.

- $\text{TRI}_{\text{Plain}}$: The first variant uses only the content encoder. During model training, the adaptive learning gates in Equation (4.8) are fixed to zero values $\boldsymbol{\sigma}_r = 0$ to exclude rating information. The learned content representations $\mathbf{h}$ are then used to predict review helpfulness.
- $\text{TRI}_{\text{Non-adaptive}}$: The second and third variants remove the adaptive learning of rating information. To this end, the gates respectively in TRI_{Add} and $\text{TRI}_{\text{Concat}}$ are set to $\boldsymbol{\sigma}_r = 1$. During training, the full amount of rating information will flow into the learned content representations $\mathbf{h}$. The final representations $\mathbf{h}'$ are then used for helpfulness prediction.
- $\text{TRI}_{\text{Raw-ratings}}$: The fourth variant downgrades rating representation from vectorized embeddings to raw values. Similar to the CM1 baseline, the ratings $\mathbf{r}$ and learned content representations $\mathbf{h}$ are concatenated to represent helpfulness.

Table 4.3: The performance of TRI variants.

Variants		D1	D2	D3	D4	D5	D6
1	$\text{TRI}_{\text{add+full rating}}$	71.97	78.23	80.04	86.80	82.90	80.00
2	$\text{TRI}_{\text{concat+full rating}}$	72.33	78.83	79.86	86.89	82.93	79.92
3	$\text{TRI}_{\text{plain}}$	70.35	77.89	78.81	85.09	81.55	78.55
4	$\text{TRI}_{\text{plain+raw ratings}}$	70.36	77.38	78.79	85.12	81.43	78.75

Three comparison tasks are designed to further validate (1) the effectiveness of the review content encoder, (2) that of the gating mechanisms used for adaptive rating learning, and (3) that of the text-rating interaction.

Effectiveness of the Review Content Encoder

$\text{TRI}_{\text{Plain}}$ is compared against CNN and EG-CNN to evaluate the effectiveness of TRI in encoding semantics. As shown in the table, $\text{TRI}_{\text{Plain}}$ is more capable of helpfulness prediction than other baseline encoders. The success of the TRI content encoder mainly lies in the gated combination utilizing both high- and low-level contextual text features. Compared with EG-CNN, however, $\text{TRI}_{\text{Plain}}$ is less effective on D1 and D2 since the two datasets have relatively higher out-of-vocabulary rates. EG-CNN tackles the issue by adopting subword information. In addition, $\text{TRI}_{\text{Plain}}$ achieves superior results to CM1 on most of the domains and even outperforms CM2 on D5 and D6. This indicates that the TRI content encoder may be able to learn deeper domain-specific semantics that is partly related to rating information.

Effectiveness of the Gating Mechanisms

TRI_{Add} and $\text{TRI}_{\text{Concat}}$ are compared against their non-adaptive counterparts to demonstrate the effectiveness of learning adaptive rating information. According to the table, the gating mechanisms improve helpfulness prediction in most cases. The comparison confirms the importance of controlling rating information flowing into review content during text-rating interaction. From a macro perspective, certain reviews may lack adequate product features for building helpfulness representations. In this case, rating information plays a complementary important role. From a micro point of view, the learned content representations encode the *n*-gram information from reviews. Different *n*-grams (e.g., "the best movie" and "the movie is") require varying degrees of rating information. The gating mechanisms handle such requirements by assigning adaptive weights to each rating dimension.

On D1 and D2, $\text{TRI}_{\text{Concat}}$ learns better review representations under a non-adaptive setting. One plausible reason is that $\text{TRI}_{\text{Concat}}$ has higher model complexity than TRI_{Add}. When adaptive rating learning is enabled, the former involves even more training parameters. For small datasets, the lack of training data may limit model performance. Nonetheless, the difference in accuracy between the two models is trivial.

Effectiveness of the Text-Rating Interaction

The four models, TRI_{Add}, $\text{TRI}_{\text{Concat}}$ and their non-adaptive counterparts, are compared against $\text{TRI}_{\text{Raw-ratings}}$ to highlight the effectiveness of the text-rating interaction used in TRI. According to the table, the four models significantly beat $\text{TRI}_{\text{Plain}}$ by about 1%–2% in accuracy, whereas the improvement in $\text{TRI}_{\text{Raw-ratings}}$ is trivial. This further confirms that TRI is more effective in capturing the relationship between review texts and star ratings. Three factors are essential to text-rating interaction. (1) Star rating vectorization allows for a larger representation capacity of rating information. (2) Decoupling the encoding of rating embeddings from that of review content maintains the influence of rating information. (3) Element alignment between content and rating vectors further provides more accurate and direct information flow.

It is worth noting that $\text{TRI}_{\text{Raw-ratings}}$ is slightly inferior to $\text{TRI}_{\text{Plain}}$ in several domains. The degradation probably results from review valence in texts incompatible with that in ratings. As discussed, the content encoder in TRI can, to a certain extent, learn latent features that are related to rating information. Since ratings are not distributed and adaptive rating learning is unavailable in $\text{TRI}_{\text{Raw-ratings}}$, attaching raw ratings to the learned content representations may introduce potential redundancy and noise that harm the model performance.

4.6.4 Comparison between the Combination Methods

Table 4.2 compares the performance between TRI_{Add} and $\text{TRI}_{\text{Concat}}$. As shown, one rating enhancement method does not consistently outperform another. The justification is the emotional homogeneity between review texts and star ratings. Recall that the last fully-connected layer in $\text{TRI}_{\text{Concat}}$ can be thought of as employing separate matrices to transform the learned content and rating representations. TRI_{Add} is a special case of $\text{TRI}_{\text{Concat}}$ in which the two matrices are shared, assuming higher homogeneity between the two sources. In domains where internal emotions in reviews texts are inadequate, $\text{TRI}_{\text{Concat}}$ may be more capable than TRI_{add} in performing text-rating interaction. Marco et al. [232] draw a similar conclusion that even using the same feature set can lead to domain-dependent performance. In their experiments on CD-related and movie-related reviews, the authors attribute similar performance to the two domains having more homogeneous products. In contrast, the electronic domain whose performance is far dif-

ferent includes many different types of products.

To further support the argument, the LIWC sentiment analysis is conducted to explore the emotional components of each domain. The analysis aims at showing the average percentage of words in reviews that possess either positive or negative emotions. As Table 4.4 reports, the three domains D1, D2, and D5, on which TRI_{Add} outperforms $\text{TRI}_{\text{Concat}}$, also possess higher ratios of emotional components. Given that the ratios are not proportional to the performance gains and threshold for emotion adequacy is unclear, the choice between TRI_{Add} and $\text{TRI}_{\text{Concat}}$ on new domains may require further domain-specific analysis. Nonetheless, TRI_{Add} is recommended since it entails less training parameters and yet yields similar performance.

Table 4.4: Average ratio of emotional words across domains.

	D1	D2	D3	D4	D5	D6
Positive Emotion (%)	7.17	4.78	3.57	4.66	4.41	4.03
Negative Emotion (%)	2.37	2.45	1.49	1.96	2.60	2.23
Sum (%)	9.54	7.22	5.05	6.62	7.01	6.25

4.6.5 Qualitative Analysis

Four qualitative analysis tasks are conducted to provide more straightforward and explainable evidence of the effectiveness of TRI. As an example, D4 is selected to investigate the learned model parameters.

Learned Document Embeddings

The first task illustrates the learned document embeddings used for helpfulness prediction. Specifically, the representations learned by TRI_{Add} and $\text{TRI}_{\text{Concat}}$ are compared against that by the TFIDF and CNN baselines. For each model, the output of the penultimate layer in Equation (4.9) and (4.10) is first computed. Dimensionality reduction via *t*-SNE [188] is then applied to obtain the 2-dimensional vector representations.

Figure 4.6 presents the predicted document embeddings after training. As shown, review representations learned by the TFIDF+SVM baseline are mixed and the least separable. The vanilla CNN framework provides improved separability to distinguish one

class from another. Still, there remain considerable overlaps between helpful and unhelpful reviews, in particular around the horizontal center. As for TRI_{Add} and $\text{TRI}_{\text{Concat}}$, different classes of reviews are further pushed to opposite directions and a clear boundary is observed, showing the effectiveness of TRI using text-rating interaction for helpfulness prediction.

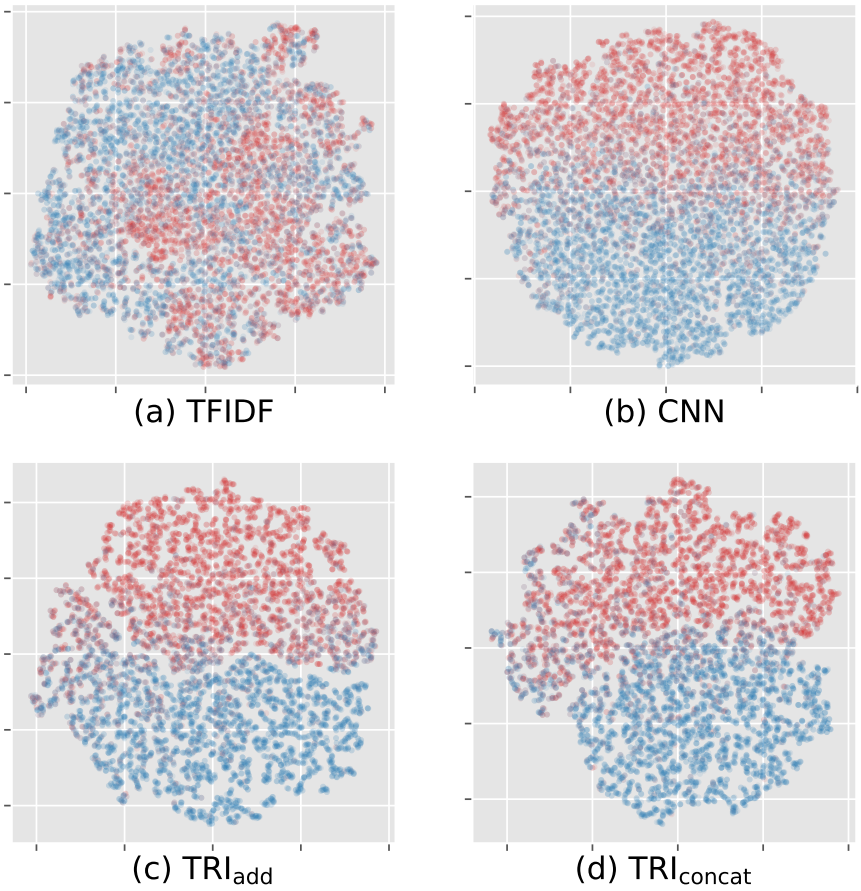

Figure 4.6: *t*-SNE projection of the document embeddings learned by (a) the TFIDF+SVM baseline, (b) the vanilla CNN framework, (c) TRI_{Add}, and (d) $\text{TRI}_{\text{Concat}}$. Blue and red points mark helpful and unhelpful reviews, respectively.

Learned Rating Embeddings

The second task studies the learned rating embeddings in TRI_{Add} and $\text{TRI}_{\text{Concat}}$ to understand the mutual relationship among different star rating levels. Following text classification conventions, the closeness between two rating embeddings $\mathbf{e}'_{r1}$ and $\mathbf{e}'_{r2}$ is computed as their cosine similarity $\frac{\mathbf{e}'_{r1} \cdot \mathbf{e}'_{r2}}{\|\mathbf{e}'_{r1}\| \|\mathbf{e}'_{r2}\|} \in [-1, 1]$. The closer a returned score is to 1 (-1),

the more similar (dissimilar) the two star levels are; 0 similarity indicates decorrelation.

Figure 4.7 illustrates the star rating similarity matrix, where the relationship between the five levels of star ratings is analyzed. Overall, the computed similarity values in TRI_{Add} and $\text{TRI}_{\text{Concat}}$ are both in compliance with the common understanding of star ratings. Take the one-star rating in TRI_{Add} as an example, its similarity with other ratings is inversely proportional to the star level. Also, a star level's previous and next neighbor possess closer similarity than the other levels. This shows that TRI can learn meaningful and effective rating embeddings.

The learned embeddings also reveal how customers perceive the meaning of star ratings. As discussed, star ratings quantitatively reflect customers' opinions and thus provide a reference sentiment for user satisfaction towards an item. While there is a consensus that one- and two-star (four- and five-star) ratings are perceived as negative (positive) experience, the perception of three-star reviews is usually ambiguous. As shown in the figure, the drastic drop in the similarity between three- and four-star rating clearly shows two polarity groups. The apparent division offers convincing evidence into rating-based review sentiment acquisition. Instead of separating reviews into positive, neutral, and negative ones, dichotomization is a more realistic solution, with one-, two-, and three-star reviews being negative, and four- and five-star positive. The reason for three-star ratings being treated as a negative emotion is highly related to the online social context. As pointed out by [172], customers tend to provide positive feedback, which diminishes the neutrality of three-star ratings.

The aforementioned findings can hopefully inspire improvement on the Likert-based rating systems used for quantifying customer satisfaction. Since customers tend to express opinions dichotomously, adjustment can be made to emphasize the positivity and negativity of customer opinions. For instance, four-point Likert scales or Yes/No questions.

Learned Adaptive Rating Ratios

The third task investigates the dependence of review content on rating information. Figure 4.8 plots the gates $\boldsymbol{\sigma}_r$ in Equation (4.8) learned by TRI_{Add}. The results of $\text{TRI}_{\text{Concat}}$ are similar to TRI_{Add} and thus skipped. Due to limited space, only the first 60 helpful and unhelpful samples in the testing set are demonstrated. Each column consists of

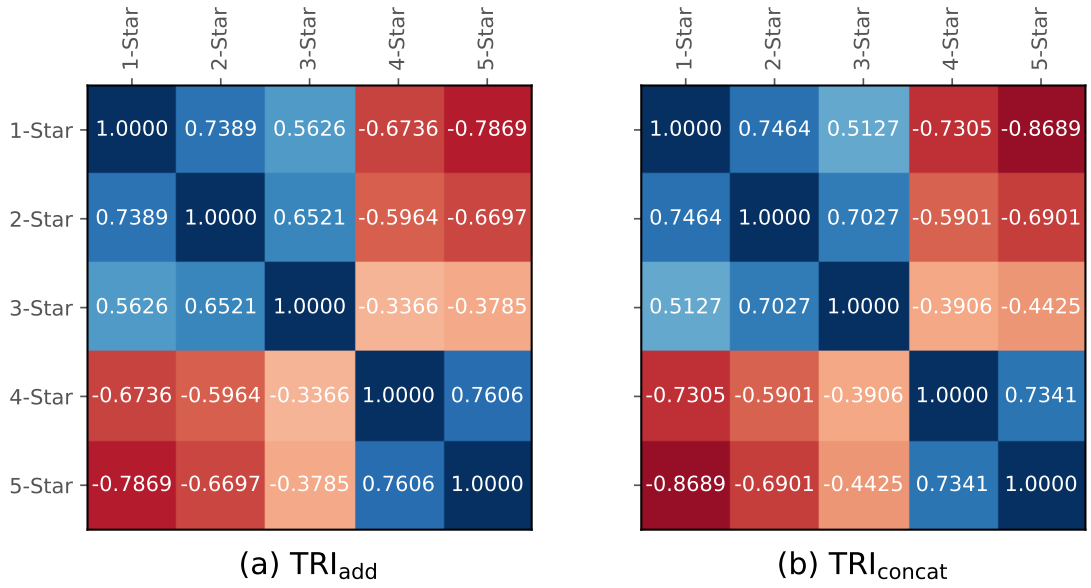

Figure 4.7: Similarity between the learned rating embedding of individual star levels. Blue (Red) color indicates positive (negative) similarity.

200 adaptively-learned ratios respectively determining the amount of rating information needed by a review's learned content representation. The ratios ranging from 0 to 1 indicate the importance of individual rating embedding dimensions.

Overall, unhelpful reviews rely higher on rating information to achieve accurate helpfulness predictions. The average gate ratio (dependency on rating information) of helpful reviews is 48.89%, whereas the number for unhelpful review is 64.32%. For some reviews, the texts per se possess comparably adequate helpfulness information, and thus less dependency on rating information is required. For instance, only a few dimensions in helpful review #14, #41, and #56 seek assistance from star ratings; unhelpful review #8 and #15 behave similarly. In contrast, rating information is in high demand in many other reviews, such as review #18, #37, and #39 in the helpful class and review #9, #10, #39, and #40 in the unhelpful class.

Several gates have high/low gate activation regardless of helpfulness categories. For example, gate #146 has a low dependency on rating information in both helpful and unhelpful reviews. Gates #27, #70, #92, and #181, however, are highly dependent on star ratings in both classes. More interestingly, some gates adapt exclusively to one type of reviews: gate #133 and #190 favor the helpful class, whereas gate #47 and #48 are far more important to unhelpful reviews.

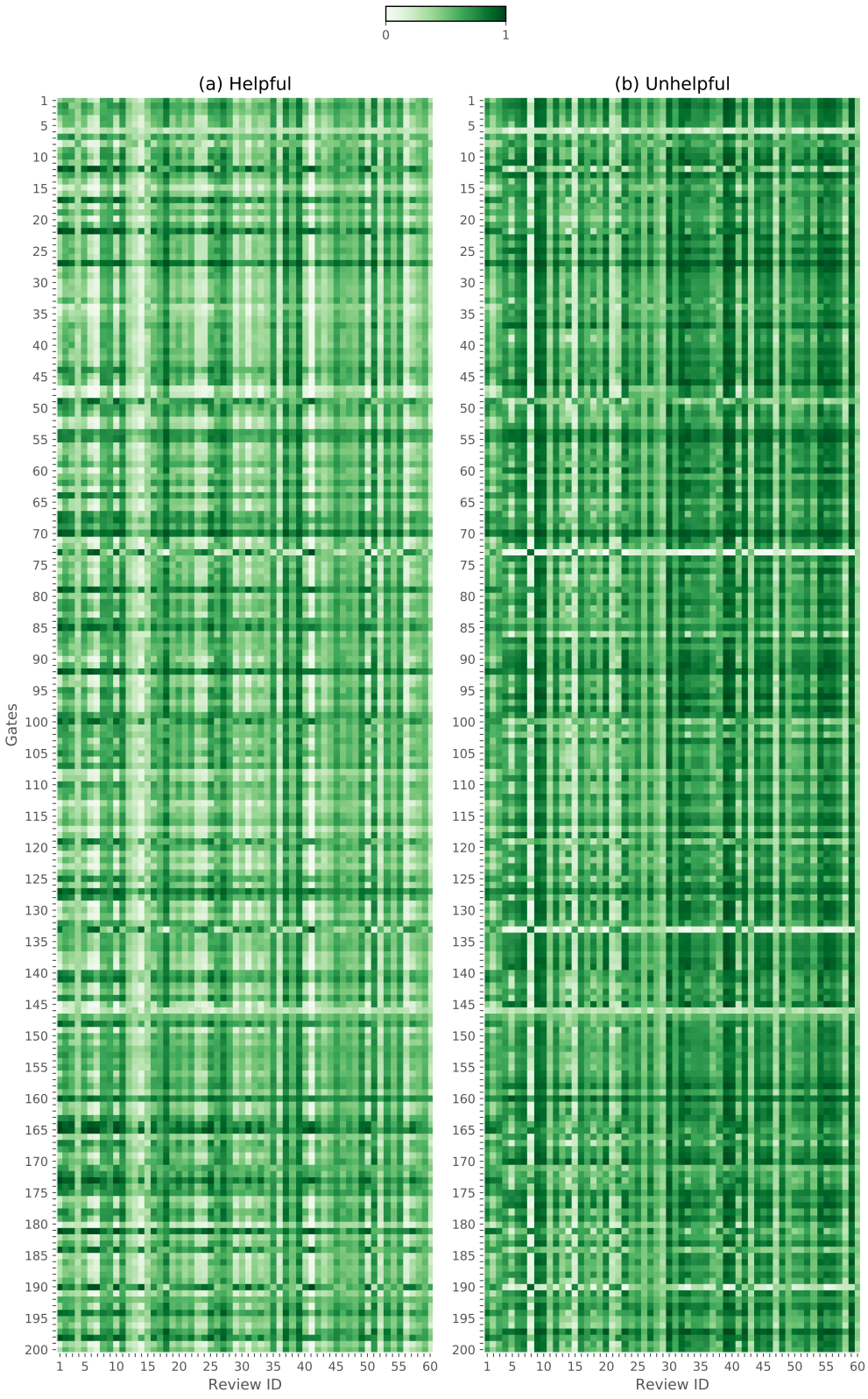

Figure 4.8: The learned amount of rating information required by texts in (a) helpful and (b) unhelpful reviews.

Case Studies

In the fourth task, the effectiveness of TRI is demonstrated with real-world examples. Table 4.5 showcases four reviews randomly chosen from the test set. The CNN baseline is used for non-rating helpfulness prediction, whereas TRI_{Add} is used for establishing the text-rating interaction. In review (a), the author appreciated the CD product overall but was dissatisfied with the price. Since readers did not expect such a comment would lead to a one-star rating, the contrast makes the review less helpful. Similarly, the mismatch between the text and rated star in review (b) confuses helpfulness perception. Review (c) marks an opposite situation where relatively negative comments were rated four stars, weakening the convincing power. Although review (d) mostly expressed negative opinions, the author suggested that the disappointment is rather regretful feelings than dissatisfaction. The four-star rating further validates and reinforces the impression, which brings high trustworthiness. The aforementioned samples provide strong evidence that text-rating interaction plays an important role in the perceptual process of review helpfulness.

4.7 Summary

This chapter has presented TRI, a deep neural architecture that learns the interaction between review texts and star ratings for helpfulness prediction. In contrast to prior work that underdevelops rating information, TRI originally (1) enlarged the encoding space of star ratings, (2) allowed for adaptive rating information learning, and (3) maintained the influence of star ratings when interacting with review texts. Extensive experiments on real-world datasets have shown the effectiveness of TRI in utilizing rating information and capturing the text-rating interaction. Ablation analysis of the TRI components further confirmed that both establishing the text-rating interaction and using adaptive rating learning are critical in improving prediction performance. Qualitative analysis of the trained parameters along with case studies offered insights and discussions for better understanding the TRI behaviors.

From a practical perspective, TRI can be hopefully integrated into existing helpfulness prediction systems. TRI takes as input review texts and star ratings for helpfulness modeling, both are standard components of a review on nearly all contemporary e-commerce platforms. Two common integration methods are available. When TRI is

Table 4.5: Examples of real-world reviews influenced by their star ratings. From left to right, each triplet indicates (1) the text-only helpfulness predicted by CNN, (2) the text-rating interactive helpfulness predicted by TRI_{Add}, and (3) the ground truth. Typos and grammatical errors are intentionally preserved.

	Review	Rating	Helpfulness
(a)	**Sleeper.** I listened to and admired Natalie Merchants voice before anyone knew who she was with 10,000 Maniacs back in the 80s. I love her voice - truly original and beautiful but this CD is a sleeper, I have to admit. And everyone - it's only $13.99 at Target (regular price). :)	★☆☆☆☆	1—0—0
(b)	**Generally Good Stuff.** An amazingly British album (which may be why I don't "get" it all). The arrangements are quite busy, and the songs and lyrics are pretty good to fantastic. I was slightly disappointed in the lack of truly "hook-y" songs - I only find myself singing a few of these the next day. "Girls & Boys", "To the End", and the punk-y "Bank Holiday" are my favorite tracks. A pretty good album, which has all the earmarks for them putting out a phenomenal one later.	★★★☆☆	1—0—0
(c)	**Loud perfection.** This is surely a fine recording, so perfect in its imperfection, a little too loud and arrogant for my taste. I don't know if it's the conductor or the orchestra, but I feel uneasy every time I listen to this powerful performance, and Volodos in spite of his great talent cannot erase that feeling.	★★★★☆	1—0—0
(d)	**Rerelease Sadly Doesn't Include Missing Videos.** When this was originally released a few years ago, I was disappointed at the omission of several videos. When I heard it was being rereleased, I hoped they would include them on the new version. Nope. That's the only reason I gave this 4 instead of 5 stars. What's there is great, but the sins of omission are unforgivable. Well, maybe if they release it a third time...	★★★★☆	0—1—1

used as a means for feature representation, the document embedding of a review learned by TRI can be used to complement that learned by an existing system. The two sets of features are then combined, upon which classification/regression algorithms are applied for final helpfulness prediction. Alternatively, TRI can be regarded as another base estimator in addition to the existing system. The final helpfulness of a review will be determined based on the predicted labels from the two (or even more) models, using max voting, weighted average, or more advanced ensemble learning techniques.

There remain several directions to be addressed. (1) Further sensitivity analysis of the TRI hyperparameters will be conducted to investigate model performance, in particular the dimensionality of word vectors, content representations, and rating embeddings.

(2) Frequent text-rating interaction patterns and domain-specific characteristics will be summarized from the trained models. Further investigation on the gating behaviors and the discrepancy (if any) in text-rating interaction between domains can hopefully offer more insights. (3) More advanced approaches will be developed to further address the interpretation of individual rating embedding dimensions and their relationship with review texts. (4) The interaction between review texts and star ratings will be constructed using more sophisticated structures such as attention mechanisms or sentence-level rating information. The diversity between reviewers in giving star ratings will also be considered. (5) The extent to which existing review characteristics (e.g., review length, text valence) affect the text-rating interaction will be studied. The characteristics will also be included in TRI for multi-characteristic interaction. (6) Inspired by existing studies working on transfer learning, the learned interactive knowledge from one domain will be applied to another. It is also interesting to build an integrated model for multi-domain helpfulness prediction.

CHAPTER 5

EXPLOITING REVIEW NEIGHBORS FOR CONTEXTUALIZED HELPFULNESS PREDICTION

Helpfulness prediction techniques have been widely used to identify and recommend high-quality online reviews to customers. Currently, the vast majority of studies assume that a review's helpfulness is self-contained. In practice, however, customers hardly process reviews independently given the sequential nature. The perceived helpfulness of a review is likely to be affected by its sequential neighbors (i.e., context), which has been largely ignored. This chapter proposes a new methodology to capture the missing interaction between reviews and their neighbors. The first end-to-end neural architecture is developed for neighbor-aware helpfulness prediction (NAP). For each review, NAP allows for three types of neighbor selection: its preceding, following, and surrounding neighbors. Four weighting schemes are designed to learn context clues from the selected neighbors. A review is then contextualized into the learned clues for neighbor-aware helpfulness prediction. NAP is evaluated on six domains of real-world online reviews against a series of state-of-the-art baselines. Extensive experiments confirm the effectiveness of NAP and the influence of sequential neighbors on a current reviews. Further hyperparameter analysis reveals three main findings. (1) On average, eight neighbors treated with uneven importance are engaged for context construction. (2) The benefit of neighbor-aware prediction mainly results from closer neighbors. (3) Equally considering up to five closest neighbors of a review can usually produce a weaker but tolerable prediction result.

5.1 Introduction

User-generated reviews play an integral part in contemporary online shopping activities. A recent survey [215] shows that 97% of customers rely on online reviews to make everyday decisions. Moreover, 85% of the customers perceive the reviews as personal recommendations. Online reviews provide new customers with opinions and experience written by previous buyers. From manufactures' perspective, online reviews also help understand consumer needs and improve product quality. Nonetheless, online reviews are uneven in quality. As a product accumulates reviews, high-quality reviews may be buried by the others of random quality. The increasing challenge requires automatic approaches for locating helpful reviews against information overload.

Table 5.1: The perceived helpfulness of a review (#3) can be affected by its neighbors (#1 and #2).

		Review
(a)	#1	This headphone is soooo cool!
	#2	Best headphone in my life. I would definitely recommend it!!!
	#3	The headphone has a fashionable appearance and the sound quality is excellent. I am surprised that it's even waterproofed.
(b)	#1	You can't find any headsets better than this.
	#2	Cheap price with good quality.
	#3	The advertisement says the headphone can last for 10 hours with full battery. Well, obviously it doesn't.

Helpfulness prediction aims to identify and recommend high-quality reviews to customers. Prior literature [224, 117, 41] has explored various features and models. One critical drawback of most existing work is the assumption that customers are unbiased and process reviews independently. In other words, a review's helpfulness is assumed to be self-contained. In practice, however, customers often read multiple reviews [215, 13] before making final decisions. Since online reviews are sequentially displayed, how and where a review is positioned [280] can potentially affect customers' perception of helpfulness. In this case, the received votes of a review may not only depend on itself but also the comparison with its the surrounding reviews.

Table 5.1 illustrates the idea with two toy examples. Assuming that customers read the reviews in order. In example (a), review #1 and #2 set a positive impression of a headphone product. Review #3 shares a similar and yet more detailed opinion, which reinforces the impression. Within the context of review #1 and #2, review #3 is seemingly more convincing and likely to receive higher helpfulness than by itself. Example (b) shows another situation where review #3 provides new information (i.e., defects of the headphone) that differs from review #1 and #2. In this case, review #3 can be more helpful due to the new information, or less helpful due to contrasting the existing impression. Both examples indicate that the perceived helpfulness of a review is not always self-contained nor independent, and the influence of a review's neighbors should be taken into account.

Different from the vast majority of prior research, this work hypothesizes that the helpfulness of a review not only depends on itself but also its neighbors. A deep neural architecture is proposed for Neighbor-Aware helpfulness Prediction (NAP). NAP first

learns representations for individual reviews. For each review, three intuitive types of review neighbors are considered: (1) preceding reviews, (2) following reviews, and (3) surroundings reviews. Four weighting schemes are then explored to construct context from the neighbor representations. During helpfulness modeling, the interaction between a review and its neighbors is captured by aggregating the contextual clues.

Note that the terms "context" and "neighbor" have been used considerably differently [224] in helpfulness prediction and pertinent fields. In most cases, context indicates information extracted from the same review as opposed to content, namely, review texts. Such information includes product metadata [97], reviewer characteristics [124], and reviewer historical voting data [97, 124, 182]. Still, reviews are treated independently and no review interaction is captured. Context can also suggest information beyond individual reviews. In [182, 204, 291], reviews are interacted via user idiosyncrasies and rater-reviewer social connections. Under this setting, users with similar preferences [186, 265] are occasionally referred to as neighbors. In [280], neighbors are defined as surrounding reviews of a given review. While the former type of neighbors have been broadly researched, the influence of the latter remains understudied.

This work targets neighbor-aware helpfulness prediction. Specifically, the helpfulness of each review is contextualized into its neighbors. Similar to [280], neighbors are clarified as adjacent reviews of a given review in a review sequence displayed to customers. The terms "neighbor-aware" and "contextualized" are henceforth used interchangeably. On the other hand, methods that only depend on information within individual reviews are called independent helpfulness prediction. More details of independent and contextualized helpfulness modeling will be discussed in Section 5.2.

To the best of our knowledge, this work offers the following contributions:

1. **End-to-end neighbor-aware helpfulness**: This work is one of the pioneer studies considering the interaction between a review and its neighbors when modeling helpfulness. Previous work majorly interacts reviews from a global perspective, using the whole review collection as context. This work instead aims at the local interaction (i.e., neighbors) among reviews. NAP also provides the first end-to-end solution for contextualized helpfulness modeling.
2. **Comprehensive contextual settings**: NAP allows for three neighbor selection and four weighting schemes for context construction. To ensure the flexibility of neighbor utilization, the four weighting schemes (each with increasing learning

parameters) construct contextual information from a various number of preceding, following, and surrounding neighbors.

3. **Extensive evaluation and analysis**: A series of experiments are conducted to evaluate the effectiveness of NAP. Hyperparameter studies are further analyzed investigate model sensitivity to discuss the trade-off between model complexity and performance. Qualitative analysis provides visualization and case studies for better understanding the model interpretation. Experimental results show NAP is effective in neighbor-aware helpfulness prediction and offer insights into utilizing neighbors for the task.

The remaining of the chapter is organized as follows. Section 5.2 surveys existing studies on independent and context-aware helpfulness prediction. Section 5.3 formalizes the problem of neighbor-aware helpfulness prediction and presents the NAP framework. Section 5.4 describes experiment settings for evaluating NAP against a series of baselines. Section 5.5 demonstrates the effectiveness of NAP, performs sensitivity analysis on contextual settings, and provides qualitative analysis on the trained models. Section 5.6 summarizes findings and discusses future research directions.

5.2 Related Work

Helpfulness prediction can either be approached in an independent or contextualized manner. The former (as most studies did) assumes that the helpfulness of a review is self-contained. The latter adopted by more recent studies considers helpfulness as an interactive function of a review and its counterparts. The following subsections survey literature on the two categories of helpfulness prediction and discuss social influence on helpfulness perception.

5.2.1 Independent Helpfulness Prediction

The vast majority of existing work predicts a review's helpfulness merely using information contained in itself. In the past decade, a large body of hand-crafted features [224, 117, 41, 75] have been carefully curated to represent the helpfulness of a review, including review text [75, 189], review metadata [313, 153], and reviewer characteristics

[48, 122]. Once chosen, the features are concatenated to represent a review and then fed into traditional machine learning algorithms for helpfulness prediction. Such methodology has the merit of easy implementation and clear interpretation due to the feature engineering nature. However, preparing effective features requires domain-specific expert knowledge, which is laborious.

Recent studies approach the task via deep learning techniques. With neural architectures, the latent representations encoding helpfulness are learned automatically, bypassing the tedious feature engineering [224] process. Currently, models developed upon convolutional neural networks (CNNs) [142, 143] and recurrent neural networks [170] such as long short-term memory networks (LSTMs) [116] and gated recurrent units (GRUs) [51] have shown to be feasible for helpfulness feature learning.

Saumya et al. [267] employ a two-layer CNN to encode review texts. Chen et al. [44] consider helpfulness modeling as a cross-domain task. To alleviate the out-of-vocabulary issue, subword information is integrated into word-level review representations. Three CNNs are separately built on top of the embeddings to transfer knowledge: one summarizes common knowledge shared across domains; the other two learn domain-specific knowledge. In another work, Chen et al. [43] extends the framework to conduct multi-domain helpfulness prediction. In addition to subword information, word embeddings are further enhanced with the distribution of product aspects [321] mentioned in reviews. In addition, gating mechanisms are adopted to learn multi-granularity text features that identify word importance in reviews.

Qu et al. [245] propose two CNN variants to combine review texts and star ratings for helpfulness prediction. The first method attaches raw star ratings as an extra dimension to the learned content representations. The second method treats each star rating as a part (the last word) of a review. Star ratings are embedded and then attached to the word embedding matrix for content representation learning. Although star embeddings enable larger encoding capacity, the current integration method largely restricts rating information from interacting with review content. Du et al. [77] cope with the issue by separating the encoding of rating embeddings from that of review content. To ensure the direct influence of star ratings on review texts, star embeddings are aligned to and then interacted with the convoluted content embeddings.

Fan et al. [85, 86] integrate rating information by formulating helpfulness prediction as a multi-task learning problem. In [85], an attention-based CNN is employed to

encode review texts. In [86], the authors model into helpfulness the semantic closeness of review texts reflecting on characteristics mentioned in the targeted product title. Two sets of bidirectional LSTMs are first used to learn separate representations for review texts and the product title. The closeness is then measured via attention mechanisms, which are used to reinforce review representations. The learned representations in both cases are then used to predict the helpfulness of a review and the accompanying star rating simultaneously.

Ma et al. [187] investigate the extent to which photos posted along with reviews influence the perceived helpfulness in the hotel industry. To this end, text representations are learned by LSTMs, whereas image representations is obtained via a pre-trained 152-layer deep residual network [112]. Both learned representations are then concatenated and fed into another LSTM to predict review helpfulness.

The independent assumption helps simplify the process of data preparation and model construction. As will be discussed, however, human helpfulness perception is more complicated and involves a variety of social biases. As such, the assumption may lead to unreliable and problematic prediction in practice. This work instead hypothesizes that a review's helpfulness depends on both itself and the context it is fit into. More specifically, the context of a review is referred to as information learned from its spatial neighbors.

5.2.2 Social Influence on Helpfulness Perception

Social influence [52, 206] has been proven to be a key part in decision making through extensive experiments [263, 211, 19, 102, 198] in psychology, economics, sociology, and human behavior analysis. The core idea of social influence is that one's decision can be affected by the presence and behavior of others [52, 206, 68, 173, 26]. Such influence also takes effect among strangers [55] and in online environments [317, 178, 63]. In the context of helpfulness perception, decision making refers to customers perusing online reviews and then determining the extent to which the reviews are helpful. Currently, the perception process is subjective varying from customers. Thus, the task is vulnerable to social influence.

Many existing studies [12, 219, 136, 244] attribute the social influence on helpfulness perception to the sequential nature of online reviews. Since reviews are sequen-

tially displayed, how a review is positioned and presented [68] to customers can affect its perceived helpfulness. Qiu et al. [244] confirm the presentation order of positive and negative reviews can influence the cognitive outcomes of readers. A line of experimental studies [278, 304, 228, 283, 203] conclude that customers are biased by past reviews when processing subsequent ones. In [280], Sipo et al. observe helpfulness voting being used as adjustment to "correct" reviews that customers believe should have a lower/higher ranking in the sequence. In consequence, helpfulness evaluation rarely takes place independently. The findings above have been adopted in star rating prediction [178, 104, 308], yet little is known how review order influences review helpfulness perception.

Recent studies further reveal the role of review order in helpfulness perception. One plausible explanation is the confirmation bias. Customers usually have their own understanding and thoughts (initial beliefs) towards products before searching. In this case, the goal of reading reviews is to gain further confirmation to support the preset expectation. When encountering a review that deviates from the expectation, customers may perceive the review as less helpful since the expressed opinions violates their initial belief. As stated in [326], more certain (uncertain) initial beliefs may lead to more (less) pronounced confirmation bias.

Another similar explanation is the anchoring effect. Unlike the confirmation bias where customers hold their own initial beliefs, the first impression [247] is formed during review perusal. According to Daomeng et al. [105], customers establish a reference frame [332] to evaluate their personal voting behavior. A relative majority opinion is learned from past reviews and compared with subsequent reviews. The majority opinion sets the initial beliefs (i.e., anchor), whereas the subsequent ones serve as new opinions. When comparing the two types of opinions [335, 246, 181, 68] (in terms of text informativeness, valence, etc.), the resulting (in)consistency [301] can affect customers' perception. Zhang et al. [335] summarize three evaluation patterns for the (in)congruent opinions using the assimilation and contrast theories.

Last but not least, review helpfulness can be explained using the information theory—whether a review provides new information. Jorge et al. [93] argue that if later reviews provide little or no new information apart from what has been described in early ones, themselves may be less helpful regardless of quality. A similar view is addressed in [295]. If words in a review are partly shared by other reviews, the review is to a certain extent predictable based on previous ones. Hence, the review is of lower

uncertainty and expected to be less helpful to a reader.

5.2.3 Contextualized Helpfulness Prediction

Few studies have attempted to integrate information beyond individual reviews into helpfulness modeling. It is worth noting again that the term "contextualized" investigates the interaction between a review and its surrounding neighbors rather than that in [182, 204, 291] modeling user idiosyncrasies and rater-reviewer social connections.

Zhou et al. [346] form an order variable to assess the impact of sequential dynamics. The authors follow [101] and first sort dynamically-ranked reviews by their time stamps. The variable then records the position of individual reviews, where those posted on the same day share an identical position. The extracted order is then used as one of the variables to construct prediction models. The same review orders are also adopted by [346, 347, 94, 93]. Alzate et al. [7] introduce three types of review orders into feature engineering: reviews in a sequence that are ranked by (i) newest review, (ii) most helpful review, and (iii) highest rating review. The authors further normalize the orders into probability variables to smooth the model interpretation.

Lu et al. [182] measure the review conformity by comparing the word distribution of a review with that of the others. The authors first vectorize reviews via a unigram language model. The overall opinion is set as the average of all review vectors related to the same item. The conformity results from the Kullback–Leibler divergence between a review representation and the overall opinion.

Hong et al. [120] measure the sentiment divergence of a review from the mainstream opinion of an item. The polarity (i.e., positive, neutral, negative) of each review is first identified based on the percentage of positive and negative words in a review. The mainstream opinion belongs to the valence that shared by the majority of reviews of the same item. The divergence between a review and the mainstream opinion is defined as their valence difference.

In [93], the authors measure the incremental information entropy of each review. The entropy is defined as the number of new words in a current review beyond that have been mentioned in the manufacturer-provided product description and in its previous reviews.

These approaches mainly suffer from three drawbacks. First, many platforms constantly update review orders as helpfulness voting evolves. Apparently, one single snapshot of reviews cannot reflect the ranking dynamics [83] over time. Therefore, most of the studies are not modeling the true order information. A possible solution to cope with the issue is to obtain multiple snapshots [7, 159, 280] of the same set of reviews, but the task is time-consuming and limited to small datasets. Deciding the time granularity is also difficult. Second, customers are assumed to be aware of the whole review collection of an item (i.e., global context) when determining a review's helpfulness. As discussed, customers only have limited patience for few reviews, and thus the assumption is hardly possible in reality. Third, most of the methods focus on peripheral cues [209] of reviews for helpfulness modeling. Features derived from review texts, which arguably contain the richest information, remain underdeveloped.

This work extends and differs from existing literature as follows. (1) A novel dataset containing six domains of online reviews is created for experimentation. The dataset is advantageous since reviews regularly uploaded by new customers are consistently ranked in reverse chronological order. (2) Deep neural techniques are employed to offer an end-to-end solution that directly learns contextualized features from review texts. (3) Local context is adopted in place of the global counterpart and constructed in a more flexible and comprehensive manner.

5.3 Neighbor-aware Prediction Networks

The research problem of neighbor-aware helpfulness prediction is formulated as a binary text classification task. Without loss of generality, let $\mathbf{S} = (S_1, S_2, \ldots, S_N)$ be an ordered list of N reviews and $\mathbf{y} = (y_1, y_2, ..., y_N)$ the corresponding helpfulness labels, where $y = 1$ is helpful and $y = 0$ unhelpful. Most existing studies oversimplify helpfulness prediction of a review $S_i (i \in [1, N])$ using the independent assumption $P(y \mid S_i; \theta)$, where θ are model parameters. Such approaches are henceforth called *independent* helpfulness prediction. NAP instead associates S_i with a context $\mathbf{T}_i$ composing reviews selected from its neighbors. The goal of NAP is to predict the probability of S_i being helpful $P(y \mid S_i, \mathbf{T_i}; \theta)$, and thus *contextualized* review helpfulness prediction.

This section presents NAP, an end-to-end deep neural architecture for the task. As illustrated in Figure 5.1, NAP consists of three learning phases. The review encoding

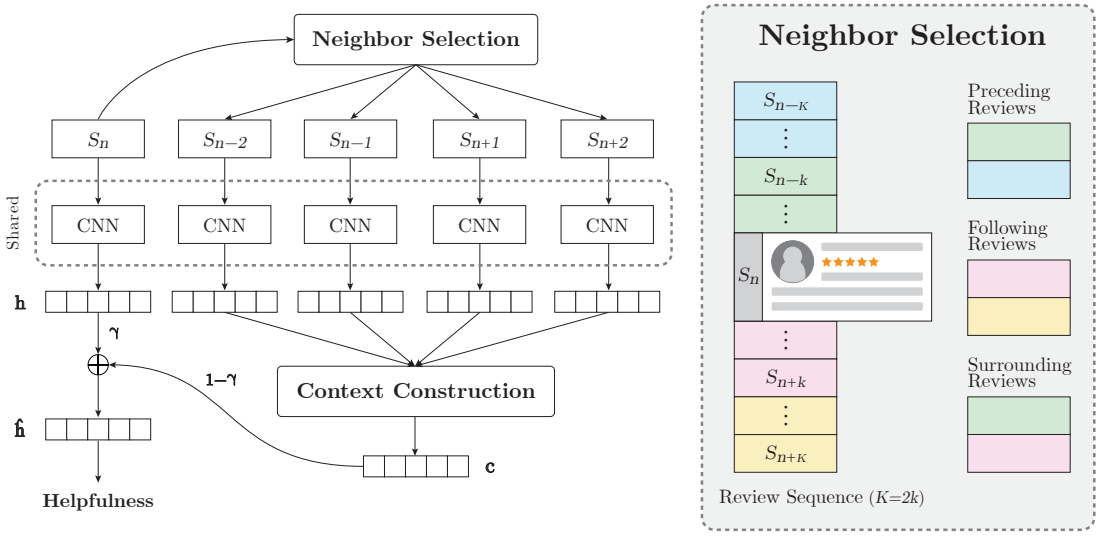

Figure 5.1: The NAP architecture. As an example, $K = 4$ surrounding reviews are selected as neighbors to construct the context of the current review S_n.

phase transforms each review S into an embedding $\mathbf{h}$. The context construction phase combines the embeddings of the associated context $\mathbf{T}_i$ into a context embedding $\mathbf{c}$. Finally, $\mathbf{h}$ and $\mathbf{c}$ are aggregated to obtained the neighbor-aware representation of S used for helpfulness prediction. The following subsections detail each model component of NAP.

5.3.1 Review Text Encoding

Let each review $S = (x_1, x_2, \ldots, x_n)$ be a sequence of n words. The vocabulary V is constructed via indexing all unique words in $\mathbf{S}$. Given an embedding lookup table $\mathbf{E} \in \mathbb{R}^{|V| \times d}$, each word $x \in V$ is associated with a d-dimensional word vector $\mathbf{e}_x \in \mathbf{E}$. Specifically, x is encoded using the one-hot encoding scheme into $\mathbf{x} \in \mathbb{R}^{|V|}$ to select the corresponding word vector $\mathbf{e}_x$. As a result, S can be represented by an embedding matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$:

$$\mathbf{e}_x = \mathbf{E}^\top \mathbf{x}, \tag{5.1}$$

$$\mathbf{X} = [\mathbf{e}_{x_1}, \mathbf{e}_{x_2}, \ldots, \mathbf{e}_{x_n}]. \tag{5.2}$$

The CNN framework proposed by Kim [142] is used to further encode the semantic meaning of review texts. Note that the goal of this chapter is neighbor-aware helpfulness prediction as a proof of concept. The focus is incorporating review neighbors as

context information instead of complex model construction. Therefore, the vanilla CNN framework is chosen to control the total number of training parameters. To learn more sophisticated review representations one could use more advanced CNN frameworks [77] and learn adaptive word- and character-level [44, 43] embeddings.

NAP employs m kernels for convolution. Each kernel is applied to a sliding window of l words over $\mathbf{X}$ to produce new features. The convoluted features are then activated using Exponential Linear Unit (ELU) [56] function:

$$\mathbf{H} = \text{ELU}(\mathbf{X} * \mathbf{W}_c + \mathbf{b}_c), \tag{5.3}$$

where the kernels $\mathbf{W}_c \in \mathbb{R}^{l \times d \times m}$ and biases $\mathbf{b}_c \in \mathbb{R}^m$ are parameters to be estimated.

The embedding of individual reviews $\mathbf{h}$ is then obtained via column-wise max pooling [60] over the feature maps:

$$\mathbf{h} = \max(\mathbf{H}). \tag{5.4}$$

5.3.2 Neighbor-aware Context Construction

NAP constructs the context of a review from its neighbors. Specifically, each review $S_i \in \mathbf{S}$ in the sequence is associated with a context $\mathbf{T}_i$ of $K = 2k, k \in \mathbb{N}^+$ reviews selected from its neighbors $\{S_j \mid j \in [i-2k, i+2k], j \neq i\}$. Three neighbor selection schemes are explored for context construction.

$$\mathbf{T}_i = \begin{cases} (S_j)_{j=i-2k}^{i-1}, & K \text{ preceding reviews,} \\ (S_j)_{j=i+1}^{i+2k}, & K \text{ following reviews,} \\ (S_j)_{j=i+k}^{i-k} \setminus S_i, & K \text{ surrounding reviews.} \end{cases} \tag{5.5}$$

The context $\mathbf{T}_i$ is regarded as reviews a user has previously read prior to the current one S_i. NAP accepts both preceding and following reviews as context because the review order in $\mathbf{S}$ does not necessarily reflect the reading order. In addition, users can vote the helpfulness of a review straight after the perusal or after reading other reviews. It can be seen that the current review S_i, by definition, can also be part of the context of other reviews.

To learn the context of a review S, the selected neighbors are mapped into embeddings and further stacked into an embedding matrix $\mathbf{C} \in \mathbb{R}^{K*m}$. The context embedding, denoted by $\mathbf{c} \in \mathbb{R}^m$, is calculated by transforming $\mathbf{C}$ via a weighting scheme

$f : \mathbb{R}^{K*m} \rightarrow \mathbb{R}^m$, $\mathbf{c} = f(\mathbf{C})$. When $K > 1$, f merges the K neighbor embeddings, which imitates customers learning the first impression $\mathbf{c}$ from past reviews $\mathbf{C}$. The weights indicate the influence of individual reviews perceived by customers. When $K = 1$, f is an identity map since one neighbor contains all information and no combination is required.

NAP introduces four weighting schemes to merge K neighbor embeddings. Each scheme is a special case of its following one, with increasing flexibility in parameter learning.

1. **Average (AVG)** The first weighting scheme borrows the idea from the neural bag-of-words model [201]. In the model, a sentence embedding results from the centroid of its constituent word counterparts, which can be thought of as a summary of the sentence. This simple model has been used in many natural language processing tasks [11, 312, 128] and proven robust and effective. Here, the context (analogous to a sentence) embedding is represented as the bag-of-reviews representation of the K neighbors (analogous to words).

$$\mathbf{c} = \frac{1}{K} \sum_{i=1}^{K} \mathbf{C}_i. \tag{5.6}$$

   The AVG scheme requires no parameters for context construction. The identical weights show equal importance of individual reviews when customers' composing their first impression towards a product.

2. **Weighted Average (WAVG)** The second weighting scheme extends AVG. In reality, user-generated reviews are uneven in quality, text valence, and sentiment intensity. Assigning separate importance for individual reviews provides higher flexibility in context construction. As such, the fixed weights in Equation (5.6) are replaced by parameters learned via an attention mechanism [249], which employs a query vector $\mathbf{u}_a \in \mathbb{R}^m$ as the learnable function:

$$z_i = \tanh(\mathbf{u}_a^\top \mathbf{C}_i), \tag{5.7}$$

$$\alpha_i = \frac{\exp(z_i)}{\sum_{j=1}^{K} \exp(z_j)}, \tag{5.8}$$

$$\mathbf{c} = \sum_{i=1}^{K} \alpha_i \mathbf{C}_i, \tag{5.9}$$

The context embedding is then obtained from the weighted average of the K review embeddings.

3. **Feature Regression (FR)** The third weighting scheme further extends WAVG. Each dimension of a review embedding suggests a certain type of latent review characteristic. During perusal, different characteristics may attract various interests. Thus, combining review embeddings on a dimension level enables more flexibility in utilizing the relationship across features. The weights are computed using a similar attention mechanism. Specifically, the context matrix $\mathbf{C}$ is first transformed into $\mathbf{Z} \in \mathbb{R}^{K*m}$ via another matrix of the same shape, followed by column-wise softmax normalization.

$$\mathbf{Z} = \tanh(\mathbf{W}_b \otimes \mathbf{C}), \tag{5.10}$$

$$\beta_{ij} = \frac{\exp(\mathbf{Z}_{ij})}{\sum_{k=1}^{K} \exp(\mathbf{Z}_{kj})}, \tag{5.11}$$

$$c_j = \sum_{k=1}^{K} \beta_{kj} \mathbf{C}_{kj}, \tag{5.12}$$

where $\mathbf{W}_b \in \mathbb{R}^{K*m}$ are learned parameters and $\otimes$ the Hadamard product. The j-th dimension c_j is then the weighted average of the same context matrix column $(\mathbf{C}_{kj})_{k=1}^{K}$. The result of c_j can also be thought of as conducting linear feature regression on $(\mathbf{C}_{kj})_{k=1}^{K}$.

4. **Spatial Feature Regression (SFR)** The fourth weighting scheme considers the interaction among neighbors. Since reviews are sequentially displayed, neighbors closer to the target review are more likely to attract higher reading priority. In addition, neighbors being read earlier may influence those later. To capture such influence, information of closer neighbors is shared with farther ones such that:

$$\hat{\mathbf{C}}_i = \begin{cases} \sum_{k=i}^{K} \mathbf{C}_k, & \text{Preceding reviews,} \\ \sum_{k=1}^{i} \mathbf{C}_k, & \text{Following reviews.} \end{cases} \tag{5.13}$$

As for surrounding reviews, the left half and right half are regarded as preceding and following reviews, respectively. The enhanced context matrix $\hat{\mathbf{C}}$ is then passed to Equations (5.10)–(5.12) in place of $\mathbf{C}$ for context construction.

5.3.3 Contextualized Helpfulness Prediction

Finally, NAP contextualizes a review within its neighbors by aggregating the embedding of a review $\mathbf{h}$ and that of its neighbors (i.e., context) $\mathbf{c}$ via linear combination:

$$\hat{\mathbf{h}} = \gamma\mathbf{h} + (1 - \gamma)\mathbf{c}. \tag{5.14}$$

Here, $\mathbf{c}$ learns the relative majority opinion [105] that can be thought of as a user's initial belief towards an item, whereas $\mathbf{h}$ serves as a new opinion. The contextualization thus learns the interaction between the initial belief and new opinion. The combination factor $\gamma \in [0, 1]$ controls the influence of neighbors on the current review. Note that setting $\gamma = 1$ stops the influence of neighbors. In this case, the helpfulness information of a review is self-contained, and thus called independent helpfulness prediction. When $\gamma = 0$, a review's helpfulness relies exclusively on its context.

The neighbor-aware representation $\hat{\mathbf{h}}$ is then forwarded into a logistic regression layer to predict the helpfulness of the current review.

$$\hat{y} = \sigma(\mathbf{W}_o^\top \hat{\mathbf{h}} + b_o). \tag{5.15}$$

NAP is trained via cross entropy minimization over M samples. The regularization on the CNN filters with weight decay λ is added to reduce the overfitting of text encoding.

$$\mathcal{L} = -\frac{1}{M}\Big[\mathbf{y}^\top \log(\hat{\mathbf{y}}) + (1 - \mathbf{y})^\top \log(1 - \hat{\mathbf{y}})\Big] + \frac{\lambda}{2}\|\mathbf{W}_c\|^2, \tag{5.16}$$

where $\hat{\mathbf{y}}$ and $\mathbf{y}$ are the predicted and actual helpfulness labels respectively.

5.4 Experiment Settings

NAP is evaluated and benchmarked against a series of baselines via extensive experiments. Section 5.4.1 describes in detail the datasets used throughout the experiments, including data collection and pre-processing. Section 5.4.2 describes the baselines using both traditional machine learning algorithms and deep learning architectures are described for performance comparison. Section 5.4.3 presents hyperparameters for training NAP and the baseline models.

5.4.1 Datasets

One critical challenge of neighbor-aware helpfulness prediction is data preparation, which requires both a review and its neighbors. Currently, many platforms dynamically rank reviews based on a set of criteria. Such mechanisms change the neighbors of a review as helpfulness voting evolves. As a result, a review's neighbors at the time of data collection only reflect a single snapshot but not its previous dynamics. One could collect multiple snapshots of the reviews through periodically tracking their ranking statistics, but the collection is expensive, time-consuming, and difficult in deciding time granularity.

This work opts for an alternative option to prepare eligible online reviews. Despite that many online platforms adopt dynamic ranking algorithms, several inherently rank reviews in reverse chronological order to provide customers with the latest user feedback of products/services. As for the latter, the static and consistent review order over time ideally compensates the necessity of multiple snapshots of reviews. In particular, two popular platforms meeting such criteria are considered: SiteJabber[1] and ConsumerAffairs[2]. Both platforms offer a wide range of categories of user-generated reviews regarding products, retailers, and companies, with SiteJabber focusing more on websites and online businesses. It is worth noting that although evaluated on chronologically-ordered reviews, NAP is also applicable to reviews with ranking dynamics provided that multiple snapshots are given.

Python scripts are compiled to crawl, extract, and store reviews from the two platforms. A total of 169,126 reviews posted prior to 29 April, 2019. The raw SiteJabber dataset consists of 60,426 reviews collected from three categories (i.e., Marketplace, Wedding Dresses, and Dating), whereas the ConsumerAffairs dataset originally contains 108,700 reviews collected from the Car Insurance, Travel Agencies, and Mortgages categories. As shown in Figure 5.2, each category (domain) of a website contains a list of reviewed items and each item consists of a list of reviews. Table 5.2 presents two review samples for each website, along with the accompanying attributes. For simplicity, the six domains are called D1, D2, and so on.

The following pre-processing steps are applied to the raw reviews to improve data quality. (1) To ensure that reviews can have adequate neighbors for context assembly,

[1] https://www.sitejabber.com/
[2] https://www.consumeraffairs.com/

Table 5.2: Example SiteJabber (top) and ConsumerAffairs (bottom) review composition. Typos and grammatical errors are intentionally preserved.

Attribute	Value
Reviewer Name	David W.
Total number of posts by the reviewer	8
Total number of votes received by the reviewer	22
Review Date	Saturday, 7 April 2018
Number of helpful votes	10
Star Rating	1
Review Title	They refused my order, didn't communicate or return my money
Review Text	I placed an order for around $200, The order went through and they took my money.after a little while I received an email that they put a hold on my order and the only way to have the order go through was to send them front and back pictures of my credit card and my passport. I refused but [...]

Attribute	Value
Reviewer Name	Justin
Reviewer Location	Heflin, Alabama
Verified Buyer	Yes
Verified Reviewer	Yes
Review Date	Sunday, 9 April 2017
Number of helpful votes	8
Rating	5
Review Text	The home loan process at Vanderbilt Mortgage was very easy going. It was also pretty fast. From the time that I went house shopping to the time that I was in my house, it took me about a month. Also, everyone I spoke to throughout the process was very informative, helpful, friendly and courteous. [...]

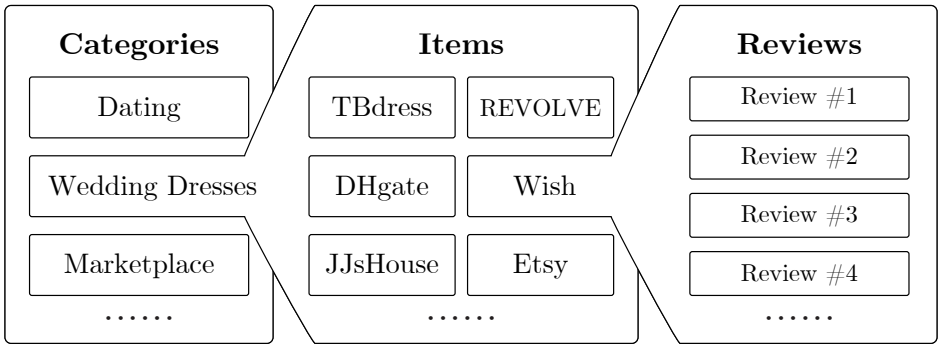

Figure 5.2: The hierarchy of the collected SiteJabber reviews. ConsumerAffairs shares the same organization.

only reviewed items with $\geq$ 100 remaining reviews are considered. (2) Each review is lowercased and tokenized in to a sequence of words, followed by minimum stopword removal that eliminates articles (i.e., a, an, and the) from the reviews. (3) Following [77], only the most frequent 30k terms are kept as vocabulary to reduce the execution cost during model training. (4) Early-posted reviews tend to receive disproportionately higher number of votes [303, 302] over later ones. To cope with the bias, reviews posted in early months that have less than 15 reviews for the same item are removed. (5) Similarly, reviews posted recently are removed due to insufficient exposure time for voting. It is worth noting that reviews with few votes are usually filtered out [258, 334] to learn more robust models. Since review order is an importance factor for correctly training NAP, this work does not perform any further removal of reviews based on the number of votes.

The pre-processed reviews are then labeled and split. Review labels are determined upon existing human assessment, namely, helpfulness votes. Following [187], a review is labelled as helpful if it receives at least two votes and unhelpful otherwise. For each domain, the constituent reviews in a reviewed item are first partitioned into three sets, using 80%, 10%, and 10% of the data respectively for training, validation, and testing. In particular, chronological split [166, 191, 182, 204, 333] is applied over randomization to preserve the review order information.

After dataset partition, review words that are numeric values are replaced by <NUM>. Similarly, mentions of names regarding the reviewed items are replaced by <ORG>. For each domain, <UNK> is used to alter out-of-vocabulary words (viz. terms that exist in the training set but are missing from validation/test set) in the reviews.

Finally, the three types of context are assembled for individual reviews within each partition following Equation (5.5). For each domain, the constructed review-neighbors pairs across reviewed items are gathered. Helpful review-neighbors pairs are randomly sampled to have the same number as unhelpful ones and vice versa to avoid class imbalance. Throughout this work, NAP and all baseline models are trained on the training set, tuned on the validation set, and evaluated on the test set serving as unseen data in reality. Table 5.3 demonstrates the simple descriptive statistics. As seen, reviews posted in ConsumerAffairs tend to be roughly twice lengthier than those in SiteJabber.

Table 5.3: Descriptive statistics of the balanced doamins after pre-processing.

Domain		#Reviews	#Words	$\frac{\#\text{Words}}{\#\text{Reviews}}$	#Sentences	$\frac{\#\text{Sentences}}{\#\text{Reviews}}$	$\frac{\#\text{Words}}{\#\text{Sentences}}$
D1	Dating	4,054	359,369	88.65	27,035	6.67	12.91
D2	Wedding Dresses	5,294	456,602	86.25	36,909	6.97	12.67
D3	Marketplace	6,964	581,456	83.49	46,222	6.64	12.31
D4	Car Insurance	2,932	398,341	135.86	27,004	9.21	14.42
D5	Travel Agencies	8,156	1,168,941	143.32	78,408	9.61	14.67
D6	Mortgages	4,602	652,223	141.73	44,955	9.77	14.13

5.4.2 Baseline Methods

The three types of neighbor-aware helpfulness prediction (i.e., preceding, following, and surrounding reviews) are compared with the independent counterpart. In NAP, independent helpfulness prediction is achieved by setting $\gamma = 1$ in Equation (5.14). NAP is also benchmarked against six state-of-the-art baselines modeling helpfulness beyond individual reviews. For simplicity, independent helpfulness prediction is henceforth denoted as **I** and the three types of neighbor-aware helpfulness prediction **I+P**, **I+F**, **I+S**, respectively.

- **I+ORD**: This baseline operationalizes three types of orders [7, 346]. The first type, denoted as $\textbf{I+ORD}_{\text{D}}$, is based on review dates. Let R be reviews of the same product sorted from the latest to the oldest, each review $r \in R$ is associated with a posted date d_r. Given a day $d' \in \{d_r \mid r \in R\}$, reviews $R_{d'} \equiv \{r \mid d_r = d'\}$ posted on the same day are shared with the same order $\left[\sum_{d<d'} N(R_d) + 1\right]^{-1}$, where $N(R_d)$ is the cardinality of R_d. Similarly, the second type $\textbf{I+ORD}_{\text{R}}$ and third type $\textbf{I+ORD}_{\text{V}}$ of orders are handled respectively by sorting reviews from highest to lowest star ratings and from the largest to smallest number of helpful votes.

- **I+CON**: This baseline measures the conformity [182] of a review $r \in R$ to reviews R of the same product. Each review $r \in R$ is first vectorized into its unigram TFIDF representation $\mathbf{u}_r$. The conformity calculates the Kullback–Leibler divergence between a review $\mathbf{u}_r$ and the overall opinion $\bar{\mathbf{u}} = \frac{1}{|R|} \sum_{r \in R} \mathbf{u}_r$.
- **I+POL**: This baseline measures the sentiment divergence [120] of a review $r \in R$ from reviews R of the same product. Each review $r \in R$ is associated with (i) a numeric polarity $p_r \in [-1, 1]$ decided by the proportion of positive and negative words in r and (ii) a categorical polarity $c_r \in \{\text{negative}, \text{neutral}, \text{positive}\}$ based on p_r. The divergence results from the absolute difference between a review p_r and the mainstream opinion $\bar{p} = \frac{1}{|R'|} \sum_{r' \in R'} p'_r$, where $R' \equiv \{r' \mid c_r = c'\}$ and c' belongs to the categorical polarity that shared by the majority of reviews in R.
- **I+ENT**: This baseline measures the incremental information entropy [93] of reviews R of the same product. Let R_n be the n-th review, $n \in \mathbb{N}^+$, $\text{vocab}(R_n)$ returns the total number of unique words occurred in $\{R_m \mid m = 1, 2, \ldots, n\}$. The entropy increment of R_n is defined as $\text{vocab}(R_n) - \text{vocab}(R_{n-1})$, which computes the increased number of unique words in R_n beyond that have been mentioned in $\{R_1, R_2, \ldots, R_{n-1}\}$.

In the baselines above, the proposed contextual information is used in conjunction with many other features, which are out of the scope of this work. To enable fair comparison, the orders extracted as per each baseline and the text embeddings $\mathbf{h}$ learned via $\mathbf{I}$ are concatenated and then fed into a feedforward layer for helpfulness prediction. NAP mainly differs from the baselines in that it locally takes neighbors of a review rather than the whole list of reviews as context.

5.4.3 Hyperparameters

The lookup table $\mathbf{E}$ is initialized with the 300-dimensional public-available GloVe word embeddings [237] and kept static during training. NAP employs $m = 100$ kernels of patch size $l = 3$ for review text encoding. Inspired by that most customers pay attention to no more than 10 reviews [13] before making purchase decisions, the number of neighbors K for context construction is chosen between 1 and 10. The combination factor γ is initially set to 0.5 to assign equal importance to both the current review and its context. The weight decay for kernel regularization is set to 5×10^{-4}.

The remaining network weights are initialized using the Glorot uniform initializer [100] and updated through stochastic gradient descent over shuffled mini-batches of size 64 using the Adam [145] update rule. During training, early stopping is applied when the validation loss has no improvement for 10 epochs.

For reproducibility, all randomization processes involved in the experiments are initialized with the same random seed. The training of each model/baseline is repeated five times to test model robustness under different random initialization.

5.5 Result Analysis and Discussions

NAP is first quantitatively evaluated via extensive experiments, followed by discussions on the effectiveness of NAP and model sensitivity to different context settings. Qualitative analysis is then conducted. Throughout the experiments, model performance is measured by classification accuracy.

5.5.1 Comparison with Baseline Methods

Table 5.4 benchmarks NAP against the baselines. The used context settings of **I+P**, **I+F**, and **I+S** are based on those yielding the highest performance. In the table, results outperforming both the independent counterpart **I** and the baselines are in italic, whereas the highest results are in bold.

In brief, NAP achieves the highest accuracy across domains and leads by approximately 1% to 5%. On average, NAP engages eight neighbors for context construction. In terms of weighting schemes, WAVG and FR are more frequently adopted than AVG and SFR. Section 5.5.3 will further investigate the context settings. In contrast, the six baselines are less robust to different domains. In the experiments, most improvements are observed on D1 and D2. On D3, D4, and D6, the introduced contextual features do not influence **I** or even diminish the performance. Over all domains, the contextual features yield less than 1% accuracy gains.

Table 5.4: The results of NAP against the baseline methods. The context settings (Weighting Scheme/#Neighbors) that produce the highest accuracy are listed below.

	D1	D2	D3	D4	D5	D6
I	86.27	70.04	81.63	72.21	67.15	69.39
I+ORD$_{\text{D}}$	86.46	70.2	80.52	71.64	67.12	69.04
I+ORD$_{\text{R}}$	86.36	70.9	81.36	72.05	67.06	69.48
I+ORD$_{\text{V}}$	86.46	70.16	80.95	71.39	67.57	69.39
I+CON	86.27	70.47	80.54	72.13	67.18	69.39
I+POL	86.89	70.66	80.16	71.56	67.54	69.13
I+ENT	86.65	70.66	80.73	72.38	66.93	69.39
I+P	*89.90*	*70.98*	*83.56*	*74.84*	*67.96*	**70.87**
	FR/9	FR/10	AVG/10	SFR/6	FR/10	WAVG/10
I+F	*90.86*	*71.17*	**83.83**	*74.59*	**68.41**	*70.43*
	SFR/10	AVG/7	FR/10	WAVG/3	FR/7	WAVG/6
I+S	**90.91**	**71.25**	*83.80*	**75.00**	*67.80*	*70.35*
	WAVG/8	WAVG/10	FR/10	FR/6	WAVG/8	WAVG/4

5.5.2 What Makes NAP Effective?

In NAP, each review is contextualized within its neighbors for helpfulness prediction. Therefore, the performance gains of NAP compared with **I** and the baselines can result from (i) the interaction between a review and its neighbors, (ii) the exclusive context learned from the neighbors, or (iii) simply an increase of review data for model training. To validate the factors that lead to the effectiveness of NAP, the following NAP variants are evaluated:

- **P/F/S**: Neighbor-only prediction using merely the context embedding **c** for helpfulness modeling, by setting $\gamma = 0$ in Equation (5.14). The three types of neighbors: preceding reviews, following reviews, and surrounding reviews, are considered.
- **I+R**: Neighbor-aware prediction where the context embedding **c** encodes the same number of K reviews randomly selected from the same domain.
- **I+N**: Neighbor-aware prediction where the context embedding **c** draws random values from a uniform distribution within the range $[0, 1]$. This variant can also be thought of as introducing noise information into independent helpfulness prediction **I**.

Neighbor-aware versus Neighbor-only

Figure 5.3 compares the neighbor-aware with neighbor-only methods to validate the role of neighbors during helpfulness prediction. As depicted, **P**, **F**, and **S** receive significantly lower performance than **I+P**, **I+F**, and **I+S** across domains, respectively. The only exception is D6 where $K = 7$ following neighbors are weighted using the WAVG scheme, which produces less than 0.1% increase in accuracy. The results strongly evidence that the effectiveness of NAP lies in an independent review interacting with its neighbors rather than either of the individuals.

Overall, both the neighbor-aware and neighbor-only methods benefit from involving more neighbors. Recall that neighbors are treated as prior knowledge to support helpfulness interpretation. Involving more neighbors helps **P**, **F**, and **S** accumulate helpfulness clues, which may include those could have been mentioned in the targeted review. As a result, the accuracy of **P**, **F**, and **S** is gaining faster as K increases and less likely to plateau. Still, the accumulated clues can hardly cover all information contained in the targeted review. This explains why the neighbor-aware methods achieve higher accuracy than the neighbor-only counterparts with far fewer neighbors. Without knowledge of the targeted review, **P**, **F**, and **S** also perform less stably across weighting schemes and neighbor types than the neighbor-aware methods.

In several cases, the neighbor-only methods show comparable predictive power to independent helpfulness prediction. On D1, for instance, the accuracy of **P**, **F**, and **S** is close to **I** at $K = 10$. On D4, neighbor-only methods outperforming **I** is observed using $K \geq 4$ reviews. This suggests that the helpfulness of a review can sometimes be approximated by the collective helpfulness of its neighbors. On the majority of occasions, however, the effectiveness of the neighbor-only methods is weak.

Lessons Learned: The performance gains of NAP mainly result from the review-neighbors interaction. Using neighbors alone, while comparable in rare cases, is not effective for helpfulness prediction.

Neighbors versus Non-neighbors

To validate whether the performance gains result from simply inputting more reviews, the neighbors used in NAP are replaced by the two types of non-neighbor context **I+N**

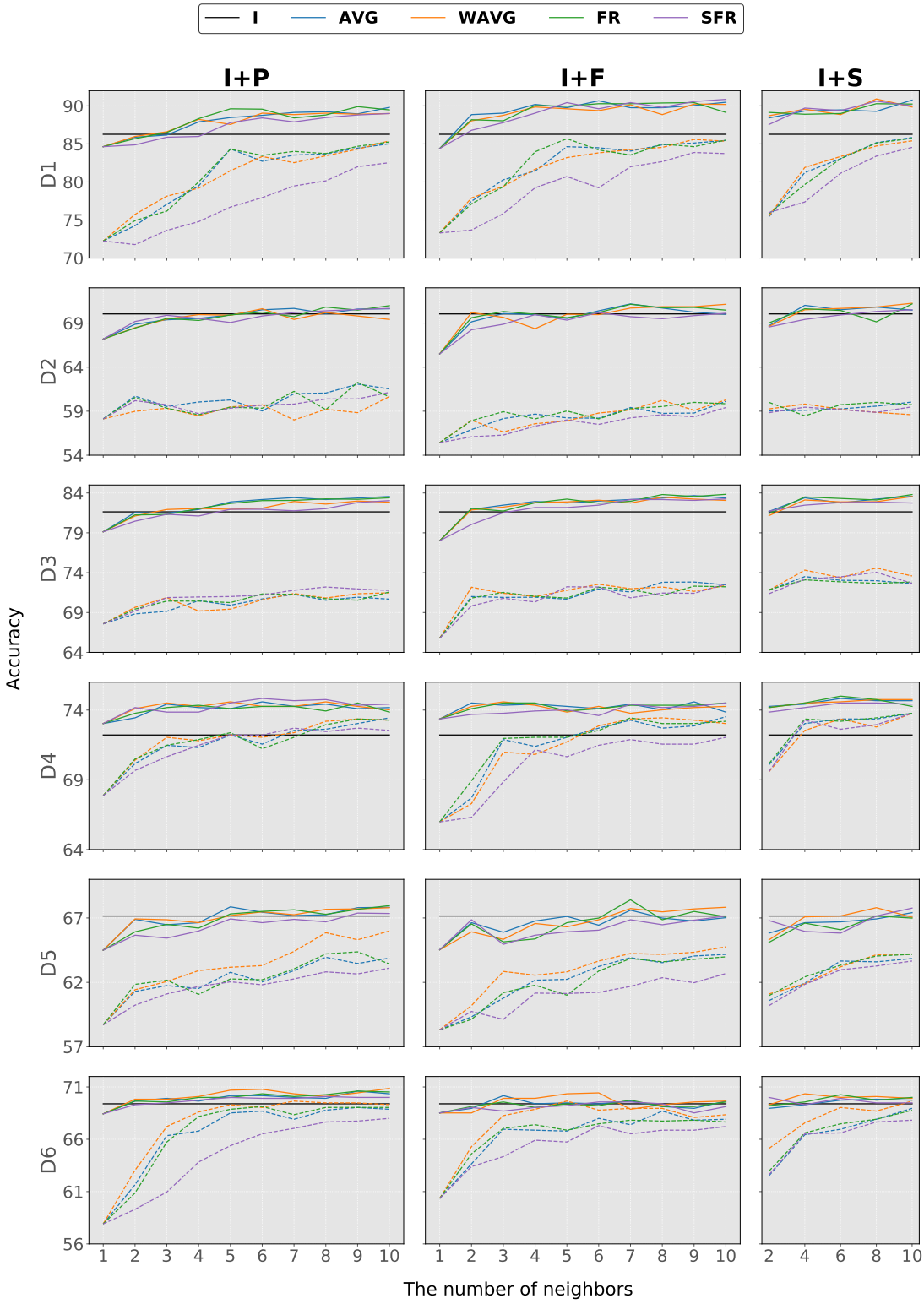

Figure 5.3: The performance of NAP on different context settings. Dotted lines are the neighbor-only counterparts of the neighbor-aware methods.

and **I+R**. Note that the SFR weighting scheme is excluded from **I+R** since random reviews do not possess spatial characteristics.

As shown in Figure 5.4, both types of non-neighbor context receive lower accuracy than **I+P**, **I+F**, **I+S**, and **I** across domains. Similar to **I+N**, **I+R** can be thought of as introducing a form of noise into **I**. Although involving more random reviews tends to improve **I+R**, the accuracy across domains, if not comparable to, is worse than **I+N**. This suggests that random reviews **R** harm **I** even more than random noise **N**. On the other hand, using **N** alone acts similarly to random guessing (50% ± 2.5%). The performance of **R** fluctuates around **N** regardless of the value of K. Compared with **P**, **F**, and **S**, simply stacking random reviews cannot accumulate helpfulness clues to form an effective context. The results prove the indispensability of using neighbors for context construction.

Lessons Learned: The effectiveness of NAP essentially relies on learning specific clues from neighbors. Simply including arbitrary reviews does not lead to performance gains.

5.5.3 Sensibility Analysis on Context Settings

Four types of NAP hyperparameters are further explored to investigate how different context settings affect the model. The hyperparameters and their possible values are listed in Table 5.5. Subsequently, the trade-off between NAP's performance and complexity is discussed.

Table 5.5: NAP context settings to be investigated.

Hyperparameters	Possible Values
The number of neighbors K	$\{i \mid i \in \mathbb{N}^+, 1 \leq i \leq 10\}$
The neighbor selection schemes	Previous, following, and surrounding neighbors
The weighting schemes	AVG, WAVG, FR, SFR
The combination factor γ	$\{\frac{i}{10} \mid i \in \mathbb{N}^+, 1 < i < 10\}$

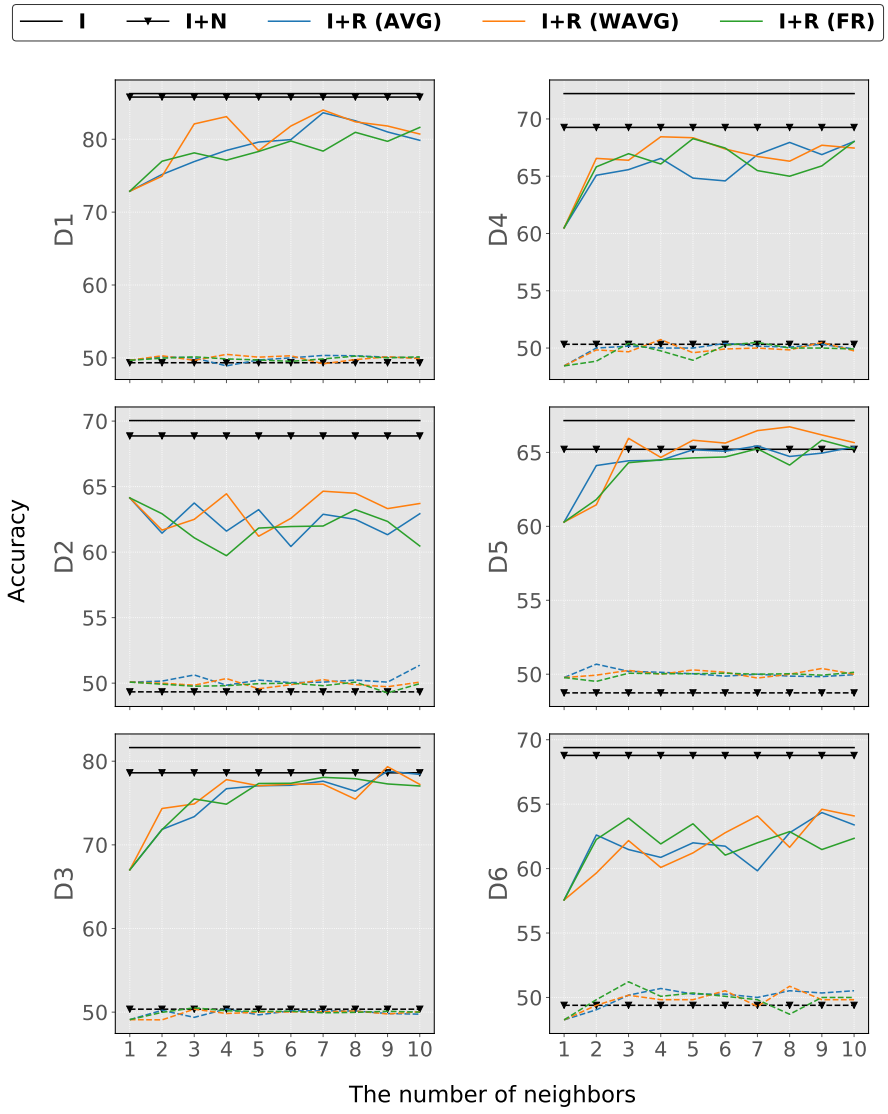

Figure 5.4: The performance of NAP using non-neighbor context. Dotted lines are the context-only counterparts.

Number of Neighbors

Figure 5.3 illustrates the relationship between the number of neighbors and model performance. As shown, NAP generally improves as K increases and then plateaus. Most domains reach the highest accuracy with a K value close to 10, but the performance gains after the first few neighbors are less than 1.5%. This confirms that neighbors closer to a review drive the bulk of the influence on customers perceiving review helpfulness. Taking all neighbor types and weighting schemes into account, NAP initially beats **I** within the first five reviews. In particular, all domains but D5 achieve so within

only the first two reviews.

Overall, NAP is inferior to **I** when learning context from extremely few neighbors. In a way analogous to **I+N**, the insufficient context information used in NAP can be thought of as introducing noise to **I**. NAP starts to improve and outperform **I** when more neighbors are involved. The additional neighbors aid consolidating contextualization by accumulating helpfulness clues. At some point, continuing to include neighbors has little influence on NAP, suggesting that the information needed for contextualization has saturated.

Neighbor Selection Schemes

The three neighbor selection schemes are compared. In particular, **I+P** is selected as the baseline to observe the change of performance from using preceding neighbors to following and surrounding ones as context. Figure 5.5 demonstrates the domain-dependent behavior of neighbor selection. On D1–D4, **I+P** generally outperform **I+F** and **I+S**, suggesting that customers rely more on preceding neighbors to determine review helpfulness. Contrarily, following and surrounding reviews are more capable on D5 and D6 of constructing context information. The performance gaps among the neighbor types are mostly within 2%.

Weighting Schemes

In a similar vein, Figure 5.6 compares the four weighting schemes by computing the performance gaps between AVG and the rest. As shown, AVG offers a robust option for learning context clues from neighbors, with the gap within 1% (2%) in most (all) cases. The highest performance (blocks in the darkest blue colors) is majorly achieved by either WAVG or FR, necessitating the use of finer-grained schemes to gather useful information from neighbors of uneven quality. Whereas modeling neighbor interactions during context construction brings less obvious improvement. In many cases, SFR receives lower accuracy than other schemes if not having comparable performance. This requires further analysis on the interaction mechanism among neighbors in future work.

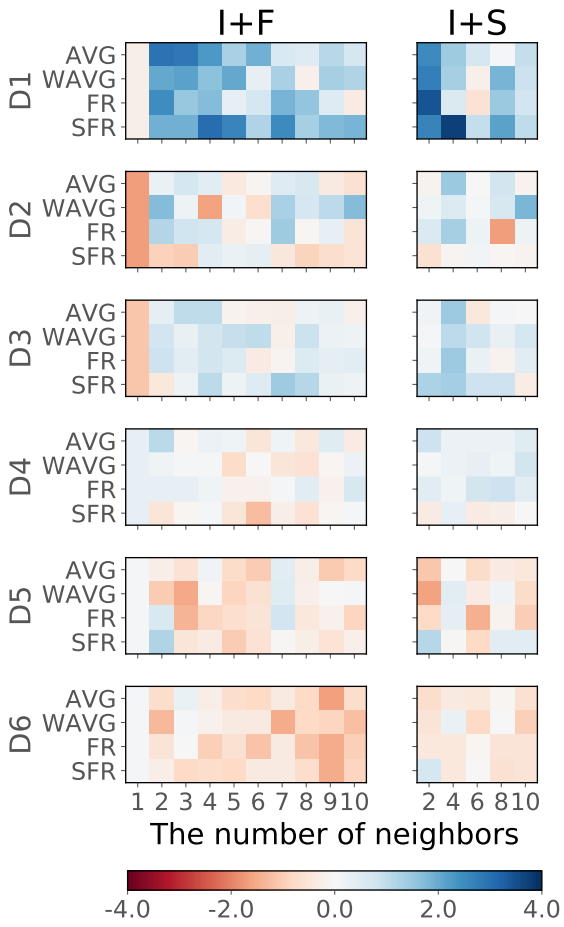

Figure 5.5: The increase/decrease in accuracy of **I+F** and **I+S** compared with **I+P**.

Combination Factor

Figure 5.7 analyzes the combination factor γ controlling the influence of neighbors on a current review during contextualization. The value of γ is varied from 0.1 to 0.9 incremented by 0.1, using the context settings mentioned in Table 5.4. The two cases $\gamma = 0$ (i.e., neighbor-only helpfulness prediction) and $\gamma = 1$ (i.e., independent helpfulness prediction) are ignored as has been reported in previous sections. Recall in Equation (5.14) that the value of γ is inversely proportional to the influence of neighbors.

As shown, the sensitivity of NAP to γ differs across domains. Overall, the performance of NAP first increases and then decreases along with γ, peaking at around $\gamma = 0.5$. This suggests neither excessive dependence on a current review or that on its neighbors facilitates contextualized helpfulness prediction. The finding further con-

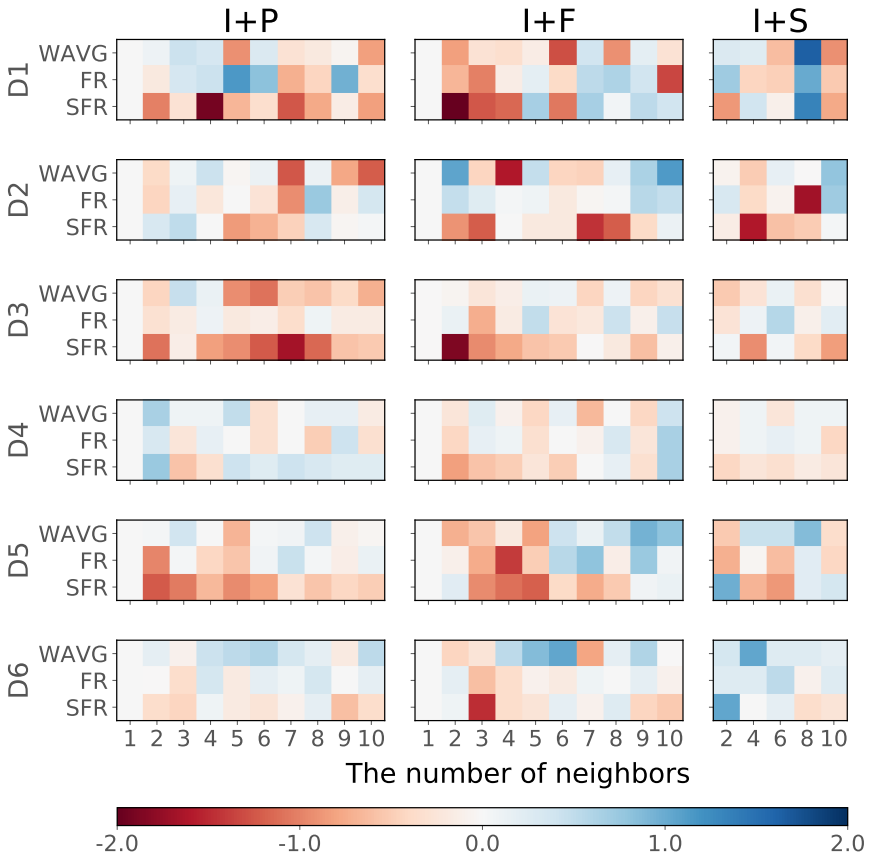

Figure 5.6: The increase/decrease in accuracy of other weighting schemes compared with AVG.

firms that the effectiveness of NAP results from the review-neighbor interaction rather than only either source.

While acting similarly across domains in $\gamma \in [0.5, 0.9]$, NAP is more sensitive to the amount of neighbor information used for helpfulness modeling in $\gamma \in [0.1, 0.5]$. Specifically, D2 and D3 show relatively high sensitivity, followed by D1 and D5, and finally D4 and D6 are comparatively less sensitive to γ. One explanation is the difference in domain-specific characteristics, for instance, the homogeneity of review opinions towards the same product.

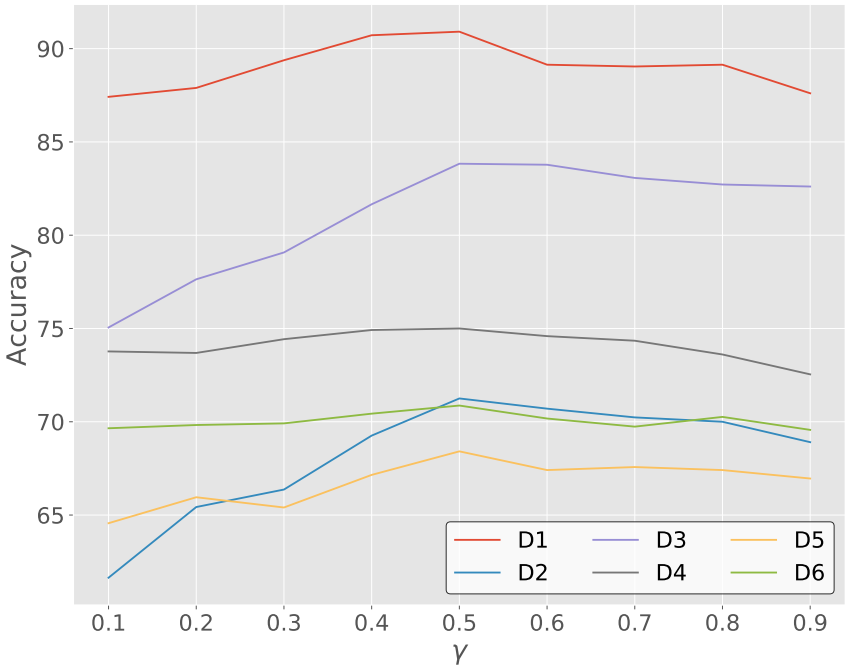

Figure 5.7: The performance of NAP on different γ values. From left to right, the influence of neighbors on a review decreases.

Trade-off between Performance and Complexity

As has been shown in Table 5.4, NAP tends to involve large number of neighbors and more flexible weighting schemes. Although reaching the highest performance, such context settings demand high computational complexity. As discussed, using excessive number of neighbors and/or overcomplicated weighting schemes does not guarantee a significant increase of accuracy. In circumstances where efficiency is emphasized, the relatively slight improvement can be traded for a faster model implementation.

This section searches for alternative NAP context settings that reduce model complexity while maintaining performance within an acceptable range. Let p be the context setting in a domain that leads to the highest accuracy q, $\hat{p}$ is a comparable alternative for p if (1) $\hat{p}$ uses smaller K values, (2) $\hat{p}$ uses simpler weighting schemes, and (3) $|\hat{q} - q| \leq \delta$. Here, $\delta \in [0, 1]$ constrains the drop of performance to be no more than 1%. Table 5.6 lists the alternative context settings ordered by δ. As shown, comparable neighbor-aware helpfulness prediction can be approached using AVG on at most five neighbors, with less than 0.72% accuracy drop. Among these alternative settings, following and surrounding reviews tend to be more effective neighbor selection schemes.

Table 5.6: Alternative context settings.

	Weighting Scheme	Neighbor Scheme	K	δ
D1	AVG	**I+F**	6	0.2392
	AVG	**I+F**	4	0.7177
D2	AVG	**I+F**	7	0.0781
	AVG	**I+S**	4	0.2344
D3	FR	**I+F**	8	0.0272
	FR	**I+S**	4	0.3261
	AVG	**I+S**	4	0.4348
D4	AVG	**I+S**	6	0.1639
	AVG	**I+F**	2	0.4918
D5	AVG	**I+P**	5	0.5502
D6	WAVG	**I+P**	6	0.0870
	WAVG	**I+P**	5	0.1739
	WAVG	**I+S**	4	0.5217
	AVG	**I+F**	3	0.6957

5.5.4 Qualitative Analysis

Two qualitatively analysis tasks are conducted to provide more straightforward and explainable evidence towards the effectiveness of NAP. As an example, D1 using the first alternative context setting (averaging the opinions of six following neighbors of a current review) in Table 5.6 is selected.

Learned Document Embeddings

The first task illustrates the learned neighbor-aware document embeddings of testing samples produced by NAP for helpfulness prediction. To this end, the output of the penultimate layer (Equation (5.14)) is computed. As for dimensionality reduction, *t*-SNE [188] is applied to obtain the corresponding 2-dimensional vector representations. Figure 5.8 presents the predicted document embeddings using neural network weights before and after model training. When the weights are initialized randomly, helpful and unhelpful samples are mixed with each other. Replacing the random weights in the embedding table $\mathbf{E}$ with those pre-trained by GloVe does not lead to significant difference. When NAP is trained, the weights of both independent and neighbor-aware helpfulness

prediction can effectively separate helpful and unhelpful samples. In particular, the latter learn better separability to distinguish helpful reviews from unhelpful ones. Therefore, the use of review neighbors as context strengthens the predictive power of helpfulness prediction.

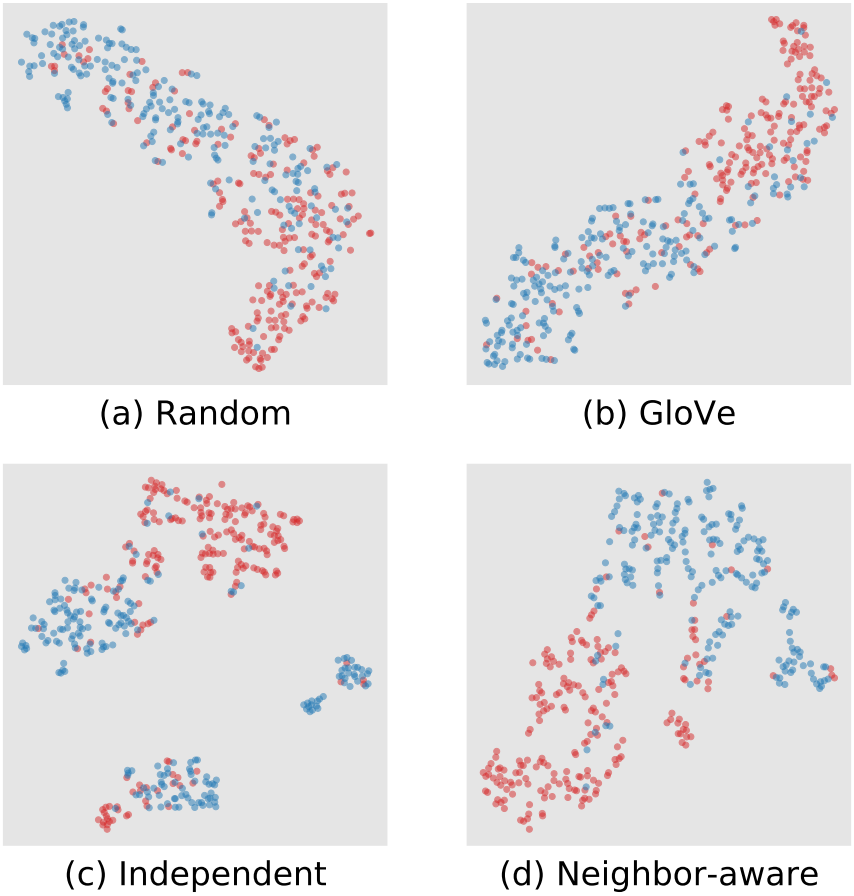

Figure 5.8: *t*-SNE projection of the learned document embeddings. Blue and red points are helpful and unhelpful reviews, respectively. (a) The model weights are initialized randomly. (b) Similar to (a) except the embedding table $\mathbf{E}$ is initialized by pre-trained GloVe embeddings. (c) The weights are trained for independent helpfulness prediction. (d) The weights are trained for neighbor-aware helpfulness prediction.

Case Studies

The second task investigates possible reasons of a current review being influenced by its neighbors in reality. Table 5.7 provides four instances from the test set, each containing a review and its neighbors, along with the predicted helpfulness and ground-truth labels.

In (a), the helpfulness of the current review per se is ambiguous. Given the context mostly mentioning a similar issue of insufficient member interaction (as underlined in the table), the current review is more trustworthy and thus wins additional helpfulness. Similarly, the neighbors in (b) aid forming an impression that the dating platform mainly suffers from pricing and customer service. Such context confirms and supports the current review, making it more helpful than it could have been if presented alone. On the contrary, (c) and (d) show a different scenario where the formed impression (overall positive) contradicts the current review's opinion (overall negative). In this case, the context weakens the perceived standalone helpfulness of the current review. The four instances above show that neighbor-aware helpfulness prediction can surpass its independent counterpart by capturing the influence brought by review neighbors.

Table 5.7: Examples of real-world reviews influenced by their neighbors. Each example contains six neighbors as the context of a current review. From left to right, each helpfulness triplet indicates (1) the predicted independent helpfulness, (2) the predicted neighbor-aware helpfulness, and (3) the ground truth. Typos and grammatical errors are intentionally preserved.

Example
(a) Helpfulness: 0–1–1
1. "I was disappointed and they took my money but no dates after 6 months of subscriptions. [...] [N]o one sent me any email or respond, no connections or dates. [...]"
2. "Sorry I joined. I joined a few weeks ago. I have seen no new people since then. The site is often down. I am not pleased and wish I had not first joined for a year."
3. "Horrible Experience. The screening process never produced results. [...] This is poor customer service and hiding behind policy when a customer is unhappy says a lot about their poor product."
4. "<ORG> [w]as the worst experience I've ever had!!! [...] [K]ept charging my credit card and never gave me any dates! It's a bunch of young kids running the office and don't have a clue what they are doing!"
5. "I found a way to close my <ORG> account. I joined this dating site 4 weeks ago and didn't like the fact I wasn't being matched with the women I put in my profile [...]"
6. "Not for those looking for real people to date. [...] [Y]et I really have had no success with any matches. It is very disappointing to be matched with at least 7 guys every day and get no response from any of them. [...]"
"Senior dating? I signed up my dad to <ORG> to see if this site works for seniors and apparently it doesn't (at least not for him due to lack of members from his town)."

continued . . .

. . . continued

(b) Helpfulness: 0–1–1

1. "Canceled account, Still charged full price. I wish I would have read more reviews before agreeing to try <ORG>. [...] Something is not right about that. :-("
2. "<ORG> - Rip OFF. [...] [E]ven when you deactivate your account, <ORG> will still charge your card. [...] [N]avigating the site and working with 'customer service' staff is a nightmare! Never again!!!"
3. "Don't Do It... Unreliable and unethical. Their customer service is horrible, the site itself is not user friendly [...] My card was charged again after having my account deactivated and their excuse was [...]"
4. "They will take your money. Be careful!!! Once you inactivate your account you will continue to get charged. <ORG> will refund only one of these charges as a 'courtesy'. [...] Customer service is completely non-existent. [...]"
5. "Stay away. Hackers and scanners have hit this site, <ORG> needs ti tighten up their security. [...] I found no customer svc support phone nimbers anywhwre on the site page."
6. "Delporable Business Practice. [...]I called to request a credit as my profile had been taken down and I thought I had terminated my account on the site. Stupid me, I thought they would deal fairly with me. [...]"

"Glitches, cumbersome site and charged full price!!!! I stupidly signed up for a year. I have met someone off line and haven't even been on the site a month. I have to pay for the entire year. Stay away from this site. [...] Horrible, horrible service!!!!!!!"

(c) Helpfulness: 1–0–0

1. "Met an awesome lady. Thanks."
2. "not too bad. better than the rest. not a hook up site for the most part."
3. "Yes and No. [...] I like the offering of options to search that <ORG> gave me. I enjoyed the formatting and the contact options. [...] I did meet someone. Thank you for a wonderful experience."
4. "God blessed me through <ORG>. [...] He is AMAZING and I truly feel God blessed me with this wonderful man. It is so good to have someone put such a smile on your face every day. I am one lucky lady!!"
5. "Met after 1 week. WE both joined about the same time. In a week we met, another week a 1st date, 5 weeks later am getting off on0line dating....hopefully for good."
6. "Met the 'Love of my Life' Great site that enabled us to meet and fall for each other!"

"Awefull. I'm embarrassed that I got on <ORG>, I should have known better. The website is awe full to maneuver. [...] I recommended NOT TO SIGN UP ON THIS WEBSITE (for your own good and $)"

continued . . .

. . . continued

(d) Helpfulness: 1–0–0

1. "I met an awesome man on <ORG>. I wasn't going to join but this handsome man kept sending me messages and I had to see what he was saying. I'm so glad I did."
2. "Located My Prince. [...] After a couple weeks of messaging, we began texting and talking on the phone. [...] We are now in a committed relationship and will be vacationing together this summer!"
3. "Hade a great time. Great site had a good experiences."
4. "It may take time but someone is there for you. Don't give up. There are many good people on this site. I have actually met a few great guys."
5. "its ok. same story as any site."
6. "Finding love quickly. This is a wonderful site-found someone with in a week."

"Total Rip Off. No matter how many miles you put, they keep sending you matches hundreds of miles away. People you contact are no longer on there. Once you cancel they use you profile forever. [...]"

5.6 Summary

This chapter has proposed NAP for neighbor-aware helpfulness prediction. NAP differs from most existing studies that assume the perceived helpfulness of a review is self-contained. NAP also differs from existing context-aware methods that learn global context from a whole sequence of reviews. Instead, NAP contextualizes a review into a small number of its sequential neighbors, which better describes the reality. In NAP, a total of 12 methods (3 neighbor selection schemes $\times$ 4 weighting schemes) were explored for context construction from neighbors. Extensive experiments on six domains of real-world reviews were conducted to validate the feasibility and effectiveness of NAP. Empirical results and qualitative analysis show that exploiting the interaction between a review and its neighbors can improve helpfulness prediction and advance the state-of-the-arts.

NAP was investigated under different context settings. Those producing the highest performance revealed that NAP engaged on average eight neighbors for context construction and considered the neighbors to be of uneven importance. The bulk of NAP's performance gains occurred in closer neighbors, whereas more distant ones had less influence. Selecting a type of neighbors for context construction, however, was domain-dependent, with following and surrounding neighbors being more favoured . Cross-domain analysis further revealed that a highest-performance context setting could be

approximated by averaging the opinions from no more than five closest neighbors of a review. The findings of this work will hopefully pave the way for future research in neighbor-aware helpfulness prediction.

There are several directions to be addressed. In the text encoding phase, more sophisticated representation methods will be employed to learn deeper semantics from review texts. As for context construction, more flexible schemes will be explored to select and aggregate neighbors. One example will be using skipped neighbors or asymmetrical surrounding neighbors. In addition, a learned rather than specified combination factor can further automate the helpfulness modeling process. Finally, further analysis on NAP will be conducted to investigate the performance gaps among domains. It is also interested to check how including even more neighbors (e.g., up to 20) will affect the performance of neighbor-aware helpfulness prediction.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

The prosperity of Web 2.0 has brought fundamental changes to contemporary shopping activities. Apart from unilateral manufacturer-provided information, customers can now rely on crowd-sourced opinions to make informed purchasing decisions. Concurrently, the number of online user-generated reviews is growing faster than before, posing new challenges to the effective utilization of reviews. To combat information overload, efficacious solutions are required to filter low-quality content and locate useful information within a plethora of online reviews.

Automatic helpfulness prediction aims to identify and recommend helpful reviews to customers. On the way towards further accurate prediction, this thesis has constructed data-driven models to overcome existing drawbacks found in prior literature. This chapter recapitulates the research problems along with the corresponding solutions and findings, followed by potential directions to be addressed in the future.

6.1 Concluding Remarks

In this thesis, the following three problems found in existing automatic helpfulness analysis have been discussed and addressed.

- Lack of evaluation standards and justification for feature curation. Helpfulness prediction has been widely studied, yet the research on this topic is still fragmented and heterogeneous. The identification of features for the task tend to be arbitrary in prior research, with a lack of justification for the decisions made. The excessive use of features without effective selection strategies can easily cause feature redundancy and irrelevance. Unclear/deficient experiment settings, ad-hoc dataset unreachability, and source code unavailability further hinder result reproduction. These issues can lead to mixed conclusions and contrasting findings, affecting the generalizability of the helpfulness prediction.
- Underdeveloped rating information and its interaction with review content. The leverage of review texts and the corresponding star ratings for helpfulness modeling has been the topic of considerable interest in the past decade. Both features have shown to be effective when used either individually or jointly. In particular,

the content-rating interaction aids in capturing the (in)consistency between the two features both quantitatively and qualitatively. Despite having a strong influence on customers perceiving review helpfulness, such mutual interaction has yet to be fully developed. Current methods either limit the text-rating interaction or the capacity of star ratings during interaction.

- Unawareness of the influence of review neighbors on helpfulness prediction. Numerous psychological studies and human behavior analyses have confirmed that social influence plays a key role in decision making. In the context of helpfulness evaluation, the perception of a review by customers has been found to be susceptible to the presence and behavior of its neighbors. At the present time, however, the vast majority of existing studies assume customers are immune to a review's surrounding context when perceiving review helpfulness. These approaches have a number of impracticable and critical drawbacks. This necessitates a more realistic methodology for contextualized helpfulness prediction that takes into account a review's neighbors.

This thesis contributes to existing literature on automatic helpfulness prediction and the research community. Effective solutions have been proposed in regards to the aforementioned problems.

- Comprehensive empirical analysis and guidelines on helpfulness feature identification and selection. Systematic literature review has been conducted to summarize, count, and rank current (i) features, (ii) models, (iii) datasets, and (iv) methodologies used in the field. Aiming for justifiable feature curation processes, the 30 most frequent features derived from review texts have been identified and grouped into five coherent categories based on their functionality. Three standard feature selection scenarios are considered to enable the effective use of the identified features. More specifically, the individual features, feature combinations within each category, and all feature combinations that lead to optimal performance have been studied. Standardized evaluation protocols have been designed to ensure the task reproducibility, model comparability, and result generalizability. Extensive experiments on six public Amazon domains show several significant findings. First, combining features from multiple categories has been shown to be the most effective scenario for accuracy improvement. Second, unigrams (either represented as conventional Bag-of-Words models or word embeddings)

yield the strongest predictive power. Third, fine-grained categorical sentiments can achieve slightly inferior accuracy to unigrams with (far) fewer feature dimensions. A comprehensive practitioners' guide to feature identification and selection has been summarized based on the findings.

- Novel paradigms of text-rating interaction and interactions between review texts and other metadata for helpfulness modeling. A new family of methods have been proposed for interacting review texts with star ratings. In particular, two deep neural network architectures have been presented for review helpfulness prediction while capturing the explicit content-rating interaction. The architecture consists of two main components: (i) the content encoder learning gated convolutional representations from review texts and (ii) the rating enhancer that incorporates rating-based valence information in an adaptive manner into the learned content representations. Model performance is evaluated via extensive experiments on six public Amazon domains against a series of state-of-the-art baselines. Ablation studies and qualitative analysis are further carried out to understand the model behaviors. The promising results in utilizing text features and rating information have led to some useful findings. First, representing star ratings as continuous embeddings makes a significant improvement on the direct use of raw numbers. Second, both the explicit content-rating interaction and adaptive rating-based valence learning are critical to improving prediction performance. More importantly, the proposed interaction mechanism inspires similar applications to other review metadata.

- Advanced review neighbor exploitation for contextualized helpfulness modeling. An original methodology has been elaborated for neighbor-aware helpfulness prediction. More specifically, a flexible deep neural framework has been introduced to take into account the surrounding context on top of the self-contained information of reviews. The framework considers a total of predefined 12 methods for neighbor context construction. The methods result from the combination of three neighbor types (i.e. preceding, following, and surrounding reviews) and four weighting schemes of different flexibility. Extensive experiments are conducted on six real-world datasets to validate framework feasibility and effectiveness, along with ablation studies and qualitative analysis. Experimental results have proven that a review's helpfulness not only depends on itself; it also depends on the interaction with its neighbors. Although on average eight review neighbors are preferred to reach max-performance prediction, the bulk improvement lies in

the closest five neighbors. Finally, customers tend to consider the influence of review neighbors differently based on their descriptive content. The selection of neighbor types is domain-specific, but overall following and surrounding neighbors are more favored.

This thesis offers valuable insights into helpful review identification and recommendation. The findings of this thesis will hopefully have theoretical and practical implications for online customers to locate and understand diverse information; and for e-commercial businesses to target and grow their user base. It is worth noting that the extracted features and trained models used within this thesis can be adapted to relevant topics (e.g. fake review detection and opinion analysis) and even broader topics such as text quality evaluation.

6.2 Future Research Directions

There remains several directions to be addressed in the future regarding the computational frameworks proposed for automatic helpfulness prediction in this thesis.

- More advanced models for helpfulness modeling along with hyperparameter tuning. The helpfulness prediction solutions outlined in this thesis have been developed upon Convolutional Neural Networks (as most deep learning studies have) to encode review content into continuous representations. In the future, more advanced content encoders will be examined to validate the effectiveness of learning deeper and more meaningful semantic relationships (e.g., ironies) among words, phrases, sentences, and reviews. These could include the attention-based architectures [85, 86] such as Hierarchical Neural Networks and the Transformer [298]. As Chen et al. [45] state, the learned attention weights can better represent reviews for user preferences and item features. Similarly, Ge et al. [96] shows that both word- and sentence-level attention are found to boost model performance due to the flexibility in selecting useful information for helpfulness modeling. Another possibility is existing pre-trained language models for contextualized sentence representation learning, such as ELMo [239], BERT [71], ERNIE [338], and their variants. Also, thorough examination is required to tune and identify the optimal hyperparameter settings [336] for the proposed frameworks.

- Further sophisticated interactions between review content and other metadata. Chapter 4 has shown the feasibility and effectiveness of interacting review content with star ratings. Despite few metadata (e.g., product types, review star ratings, and review length) being chosen as moderating factors, most of the remaining are not (sufficiently) researched. As Hu et al. [122] state, existing studies tend to assume that the helpfulness-related features interact little or not at all with each other. In practice, however, some determinants are likely to have interaction effects. For example, the inconsistency between the star class of a hotel [305] and its received review star rating may affect the perceived review helpfulness. It would be interesting to know the interaction effect of different types of review characteristics on review helpfulness, following methods similar to those described in Chapter 4. For instance, the spatial characteristics of different time granularity (e.g., seasonality [93]) may affect other factors in perceiving review helpfulness. Given the increasing prevalence of mobile devices, it is also worth taking the corresponding website layouts into account. The various availability and position of review attributes may affect the understanding a consumer's understanding of helpfulness as they are perusing reviews. Additionally, the interaction between multi-modal data [18] that utilizes natural language and computer vision could help achieve more comprehensive helpfulness. The work of Ma et al. [187] is an example. This works uses both hotel review texts and the user-uploaded images for helpfulness prediction. Finally, mutual interaction can also be extended to the interaction among multiple types of metadata.
- The exploration of data augmentation techniques. Data augmentation refers to operations on training data prior to model construction that aim to either improve data quality or produce additional training data. Data augmentation has been widely used in image processing [112], such as geometric transformations (e.g., cropping, rotating, and flipping) and color distortions. Recently, data augmentation has shown some success in NLP and relevant fields [310], and thus could be adopted to strengthen helpfulness prediction. Potential usage includes (i) enhancing unstructured raw textual reviews with structured knowledge via taxonomy learning [35]; knowledge graphs [234], argumentative structures [114, 171], and existing linguistic resources (e.g., thesauruses, ontologies, and lexicons); (ii) advanced task-specific pre-processing such as word transformations (e.g., synonym/antonym replacement) and noise interference (e.g., random word/phrase masking and swapping); (iii) post-processing on the learned word and document

representations [210]). For example, Maroun et al. [191] suspect that customers may not (carefully) read lengthy reviews because it takes extra time and effort. Therefore, skipping certain reviews or words/sentences of a review during model training may be closer to how customers actually process information when reading reviews.

- Managing social biases involved in helpfulness modeling. Existing literature [68, 172, 303] has revealed a number of helpfulness evaluation biases pertinent to social psychology and praxiology. Such biases are mainly rooted in different platform layouts and user navigation preferences. The diversity of cultural background can also affect the ways that users give, receive, and understand the sentiments [348] embedded in review opinions. Following the voting process mentioned in [224], customers can vote on whether a review is helpful based on the provided review information. As time elapses and the review accumulates votes, the corresponding review information can change dynamically. Such dynamics will in turn affect the later customers' voting behaviors (these behaviors have been partly addressed in Chapter 5). Additionally, most platforms provide options to order reviews differently, which cannot be reflected in the final voting data. Future studies need to disentangle such reciprocal influence posed by social biases. The most ideal way of doing this is to collect multiple snapshots data [280, 159, 7] at the price of long-term data collection. Alternatively, one can collect reviews that are constantly displayed in (reverse) chronological order, which strictly limits the available data sources. Given several studies having released human annotation results, a more practical method is to train models to learn the transformation between voting data and human-annotated helpfulness. The trained model can then be used to predict the "actual" helpfulness of more reviews. Following the footsteps of in many existing studies, future directions should also utilize the findings and theories from previous psychological and behavioral analysis when constructing helpfulness evaluation models.

In a more general context, the mechanism of human helpfulness assessment processes on online reviews (to which all current helpfulness modeling methods are attempting to approach) are yet to be completely revealed. In the future deeper and more thorough analysis is required to further understand, dissect, and clarify the involvement of different types of participants [208] into the whole life cycle of electronic word-of-mouth. The followings are some promising future directions for exploration:

- Towards personalized helpfulness prediction. For simplicity, most existing studies assume that all customers share a global standard to rate reviews fairly and unanimously. Reviewers can provide content of random quality due to factors such as different education backgrounds. Given this, it is natural to also consider that raters can also rate reviews in diverse ways. As Fan et al. [86] state, customers may focus on different review aspects based on their needs. Moghaddam et al. [204] confirm that the evaluation of review helpfulness varies in (groups of) users rather than being unified and standardized. Hence, the global assumption should be extended to personalized helpfulness prediction to suit more realistic situations. Several studies [45, 291, 96, 204, 291, 191] offer good starting points. Within these studies, personalization is fulfilled by taking into account review information (content- and context-based features), targeted item profiles, author profiles, rater profiles, and follower profiles (where possible). To more accurately pinpoint user-specific information, one can borrow methodologies and harvest techniques used in closely related areas, such as content-based recommendation [342] and aspect-based sentiment analysis [240]. In the future, a four-dimensional interactive framework will be designed to assist in helpfulness prediction. Focused on review content, the framework will integrate the profile of targeted items, reviews, reviewers, and raters, which are latent factors to be estimated. The learned results can be analyzed for mining customers' behavioral patterns and regular changes of preferences, categorized into personality groups, and further utilized for personalized recommendation.
- Towards versatile helpfulness prediction. Current helpfulness prediction tasks require labelled data for model training. In practice, the number of available online reviews vary hugely between items/domains/languages. The scarcity of training data may fail to truly reflect the feature distribution of helpfulness and may diminish model performance. Deep learning models are particularly in need of training data due to their data-hungry nature. On the other hand, training on moderate and large datasets demand expensive labelling efforts. In prior literature, review helpfulness is predominantly labelled in an automatic manner using the number of (helpful) votes received by reviews. Such voting data is, however, likely biased and can differ variously from human-annotated helpfulness [172, 322]. To this end, future work should pursue end-to-end models with an emphasis on higher model versatility and less data specificity dependency. Transfer learning using multi-task, multi-channel, and multi-input architectures [146] can be adopted to

learn more robust features for helpfulness modeling. There are several circumstances [322, 50] in which out-domain models can achieve similar but inferior performance than in-domain ones. Such a phenomenon suggests the existence of general knowledge [286] shared across data sources (e.g. common sense) in addition to source-specific knowledge. As such, low-resource data can be enhanced by general knowledge, taking advantages from less relevant and irrelevant data sources. Cross-lingual [259] and cross-domain [44, 43] helpfulness prediction are standard practice. In a practical context, reviews of an item in one online platform can also leveraged for another to increase the size of training data. Finally, it is preferable to train models on large-scale unlabelled data using few or no annotated reviews, such as semi-supervised [343] and unsupervised [295] methods.

- Towards intelligent helpfulness prediction. The interpretability of helpfulness has attracted increasing attention from the research community. This is due to the necessity of distilling deeper and more meaningful knowledge instead of merely improving model performance. For neural architectures using "black box magic" [260], finding explainable answers [99] to the high model performance is even more important. Current solutions [43, 322] mainly aim to showcase the learned importance of features, such as words/phrases and lexicon dimensions. The importance is computed via (i) regression weights, (ii) attention mechanism, (iii) gating mechanism, and (iv) pooling operations. These kinds of shallow explanations are, however, still far from understanding the logic behind the human voting process. Future models should raise interpretable helpfulness prediction to a higher level to gain clearer insights. More understandable schemes (e.g. functionalities, hierarchies, and visual cues) can be conceived to interpret the learned features from different abstract levels. Rationalizing neural predictions [163] is yet another remedy for acquiring justifiable information. The interpretations can be further combined with state-of-the-art natural language understanding and generation techniques to develop more intelligent models for helpfulness analysis. One application is comprehensive review summarization [8] over massive online reviews. For example, an ideal product summary could comprise bullet points describing intensity-based pros and cons, along with case studies and reasons for the opinions, rather than simply excerpting information pieces from existing review sentences. Further, the development of domain-specific Dialogue Systems [46] and Question Answering systems [115] is desired.

CHAPTER 7
COMPLETE DESCRIPTIVE STATISTICS

Table 7.1 shows the full descriptive statistics of the data used throughout the thesis. In particular, the Amazon 5-Core dataset is used in Chapter 3 and Chapter 4, whereas the SiteJabber and ConsumerAffairs data are used in Chapter 5. The complete list of statistics includes (1) the number of reviews in the training, validation, and test set, (2) vocabulary size, (3) the out-of-vocabulary rate in both validation and test set, and (4) the length statistics (i.e., the minimum, maximum, mean, median, standard deviation, and sum) of word count, sentence count, and the average number of words per sentence of reviews in individual domains.

Table 7.1: Full descriptive statistics of all the domains.

Dataset		**Amazon 5-Core**					
Domain		D1	D2	D3	D4	D5	D6
#Reviews	All	20,416	23,100	33,962	105,934	164,052	306,430
	Training	16,332	18,480	27,168	84,746	131,240	245,144
	Validation	2,042	2,310	3,396	10,594	16,406	30,642
	Test	2,042	2,310	3,398	10,594	16,406	30,644
#Vocab		28,141	91,671	96,675	246,485	349,832	483,404
Validation OOV	<NUM>	0.93%	1.10%	1.80%	0.91%	0.78%	0.57%
	<UNK>	1.67%	0.88%	0.91%	0.81%	0.67%	0.57%
Test OOV	<NUM>	0.93%	1.16%	1.73%	0.90%	0.80%	0.56%
	<UNK>	1.76%	0.89%	1.05%	0.80%	0.69%	0.57%
#Words	Min.	7	2	2	4	3	3
	Max.	1,641	5,531	5,143	4,813	5,234	5,436
	Mean	59.02	333.96	250.75	226.00	256.95	242.34
	Median	37	215	158	169	184	169
	Std.	70.68	368.80	302.59	205.69	249.56	254.05
	Sum	1,204,921	7,714,545	8,515,804	23,941,259	42,152,922	74,261,016
#Sentences	Min.	1	1	1	1	1	1
	Max.	89	396	348	565	388	661
	Mean	5.20	20.29	15.80	13.80	15.24	14.31
	Median	4	14	11	11	11	10
	Std.	4.29	20.98	17.08	12.04	13.49	13.63
	Sum	106,242	468,771	536,704	1,461,680	2,500,454	4,384,372
$\frac{\#\text{Words}}{\#\text{Sentences}}$	Min.	2.22	2.00	1.76	1.38	1.45	1.05
	Max.	191.00	260.50	324.00	531.00	630.00	542.00

	Mean	11.39	16.48	15.52	16.57	16.72	16.28
	Median	10.00	15.22	14.39	15.40	15.50	15.61
	Std.	6.44	8.76	8.40	8.53	9.27	6.70
Dataset		**SiteJabber**			**ConsumerAffairs**		
Domain		D1	D2	D3	D4	D5	D6
#Reviews	All	4,054	5,294	6,964	2,932	8,156	4,602
	Training	3,308	4,270	5,616	2,252	7,112	4,234
	Validation	328	512	612	436	426	138
	Test	418	512	736	244	618	230
#Vocab		11,381	11,492	15,836	9,118	17,778	11,932
Validation OOV	<NUM>	1.02%	1.01%	1.60%	1.41%	1.75%	1.86%
	<UNK>	3.31%	1.81%	2.56%	2.04%	1.00%	0.99%
Test OOV	<NUM>	1.01%	1.15%	1.22%	1.67%	1.77%	1.53%
	<UNK>	2.55%	1.79%	2.87%	1.84%	1.07%	1.26%
#Words	Min.	2	2	2	11	3	3
	Max.	9,403	2,857	1,779	2,617	2,453	2,462
	Mean	88.65	86.25	83.49	135.86	143.32	141.73
	Median	55	62	55	92	105	101
	Std.	194.58	89.08	99.67	150.65	136.30	142.35
	Sum	359,369	456,602	581,456	398,341	1,168,941	652,223
#Sentences	Min.	1	1	1	1	1	1
	Max.	324	154	95	184	96	187
	Mean	6.67	6.97	6.64	9.21	9.61	9.77
	Median	5	6	5	7	8	7
	Std.	9.34	5.53	5.80	8.56	7.78	8.85
	Sum	27,035	36,909	46,222	27,004	78,408	44,955
$\frac{\#\text{Words}}{\#\text{Sentences}}$	Min.	1.00	1.00	1.00	1.50	2.50	2.00
	Max.	201.50	186.50	200.00	42.50	65.00	95.00
	Mean	12.91	12.67	12.31	14.42	14.67	14.13
	Median	11.33	11.17	10.73	13.77	14.00	13.56
	Std.	9.23	8.23	9.24	4.73	5.32	5.04

CHAPTER 8
SANITY CHECK OF DOMAIN-SPECIFIC EMBEDDINGS

Table 8.1 and 8.2 depict the sanity check on general and domain-specific terms, respectively. As shown in the tables, the three types of pre-trained word embedding models are capable of returning meaningful results across domains. One merit of word embeddings is to identify informal writing, such as the term "bad" ("b-a-d", "baad", "baaad") and "good" ("goood", "goog", "goos"). Another merit of word embeddings is that misspelled words can be associated with their correct counterparts, for example, "convenient" has high similarity to "convient", "convienent", "convienient", etc. and "quality" to "quaility", "qaulity", "qality", etc.

At a glance, the retrieved similar entities trained on domains reflect more specific information. Take the term "price" as an example, the most similar words are forms of the lemma, relevant concepts, and various numeric values indicating product prices. Some domain-specific word embedding also demonstrate the idiosyncrasy of shallow reasoning. For instance, the returning entities of the term "inconvenient" suggest superfluous (D1), cheap-looking (D4), and unreadable (D4) characteristics might lead to product inconvenience. Similarly, the term "convenient" produces compact (D1), cost-effective (D4), biodegradable (D5), and portable (D6), which is insightful.

In addition, domain-specific embeddings can capture deeper semantic relationship between words/phrases and the targeted domains. For example, in D1 (Apps for Android) the most similar words to the term "quality" describe the "resolution", "clarity", and "presentation" of mobil applications. In D2 (Video Games), the term "quality" refers to the "workmanship", "craftsmanship", and "comfortability" of gaming products. Many similar entities of the term "discount" reveal retailer names such as costco (D3), woolworths (D4), walmart (D4), fye (D5), walgreens (D5), and bestbuy (d6), which are missing in SGNS and GV. Thus, domain-specific embeddings better reflect specifics than the two general purpose counters.

Table 8.1: Sanity check on general terms.

	good	bad	convenient	inconvenient
D1	great (0.7450)	shabby (0.6456)	convienent (0.7603)	cumbersome (0.6234)
	decent (0.7163)	terrible (0.5714)	convient (0.7599)	irritating (0.6220)
	nice (0.6420)	horrible (0.5701)	convienient (0.6874)	annoying (0.6177)
	goog (0.6377)	good (0.5651)	useful (0.6607)	nuisance (0.6103)
	game.good (0.6309)	ify (0.5490)	compact (0.6584)	disconcerting (0.6068)
	agood (0.6218)	floaty (0.5485)	handy (0.6456)	detrimental (0.5856)
	excellent (0.6119)	boringgg (0.5418)	reliable (0.6451)	forgivable (0.5841)

	chanllenging (0.6091)	allright (0.5416)	efficient (0.6371)	superfluous (0.5839)
	goood (0.6070)	isent (0.5392)	convinient (0.6333)	balky (0.5822)
	ggreat (0.6053)	judt (0.5377)	dependable (0.6070)	aggravating (0.5768)
D2	great (0.7680)	terrible (0.6297)	useful (0.6168)	annoying (0.6109)
	decent (0.7596)	baaad (0.6211)	compact (0.6116)	problematic (0.5946)
	good.the (0.7034)	horrible (0.6109)	reliable (0.6060)	awkward (0.5908)
	excellent (0.6810)	good (0.5942)	convienient (0.6057)	cumbersome (0.5816)
	desent (0.6414)	not-so-good (0.5704)	practical (0.6052)	irritating (0.5752)
	nice (0.6407)	neccessarily (0.5607)	convient (0.6008)	convenient (0.5736)
	goog (0.6324)	repeatitive (0.5487)	convinient (0.6000)	frustrating (0.5687)
	gr8 (0.6304)	postive (0.5452)	affordable (0.5855)	flaky (0.5686)
	grat (0.6236)	speller (0.5404)	comfortable (0.5840)	bothersome (0.5643)
	pritty (0.6219)	terribad (0.5324)	efficient (0.5750)	finnicky (0.5618)
D3	decent (0.7787)	terrible (0.6396)	convienient (0.8016)	awkward (0.7645)
	great (0.7278)	shabby (0.6138)	convienent (0.7662)	annoying (0.7203)
	excellent (0.7127)	horrible (0.6119)	useful (0.7322)	cumbersome (0.7166)
	descent (0.7060)	omen (0.5880)	practical (0.7249)	impractical (0.7070)
	goos (0.6744)	baaad (0.5875)	convient (0.7147)	frustrating (0.6991)
	nice (0.6513)	awful (0.5616)	convinient (0.7069)	troublesome (0.6951)
	qood (0.6437)	good (0.5611)	convent (0.7063)	aggravating (0.6921)
	fitts (0.6348)	shaby (0.5538)	versatile (0.6992)	unhandy (0.6911)
	goood (0.6347)	ggod (0.5363)	handy (0.6960)	irritating (0.6651)
	nive (0.6281)	poor (0.5206)	practicle (0.6273)	nuisance (0.6639)
D4	great (0.7201)	terrible (0.6810)	affordable (0.6411)	inconveniently (0.6403)
	decent (0.7057)	horrible (0.6276)	inexpensive (0.6133)	typographical (0.6313)
	ggod (0.6863)	abad (0.6227)	cost-effective (0.6129)	cheap-looking (0.5955)
	goodbut (0.6736)	half-bad (0.6206)	multiple-disc (0.5868)	unreadable (0.5942)
	good.the (0.6722)	necessarilly (0.6176)	reasonably-priced (0.5817)	qc (0.5937)
	good.buy (0.6610)	nessecarily (0.6062)	expensive (0.5708)	pull-out (0.5908)
	cd.good (0.6560)	shabby (0.5909)	slimline (0.5673)	impractical (0.5891)
	good.this (0.6431)	necesarrily (0.5883)	cd-sized (0.5544)	12x12 (0.5830)
	perty (0.6429)	_bad_ (0.5847)	inconvenient (0.5528)	six-panel (0.5828)
	good.and (0.6426)	terible (0.5795)	too-small (0.5520)	opendisc (0.5811)
D5	great (0.7819)	terrible (0.6727)	convienent (0.6458)	gore's (0.5621)
	decent (0.7301)	horrible (0.6494)	convienient (0.6327)	half-truth (0.5480)
	goog (0.7224)	thebad (0.6402)	conveniently (0.5407)	guggenheim's (0.5441)
	enteresting (0.7199)	awful (0.6219)	contrived (0.5361)	thruth (0.5322)
	excellent (0.6956)	semi-bad (0.6038)	portability (0.5224)	peer-review (0.5256)
	goos (0.6945)	tv-ish (0.5990)	flimsy (0.5216)	companion-piece (0.5097)
	goodand (0.6942)	good (0.5978)	inexpensive (0.5212)	fahrenhype (0.5037)
	prettygood (0.6797)	baaad (0.5928)	frustrating (0.5202)	verifiable (0.5017)
	movie.good (0.6784)	lousy (0.5749)	cumbersome (0.5198)	c2k (0.4978)
	excellant (0.6775)	b-a-d (0.5740)	biodegradable (0.5103)	quaeda (0.4964)
D6	great (0.8184)	semi-bad (0.6409)	convienent (0.7102)	inconvienient (0.6121)
	excellent (0.6981)	good (0.6264)	convient (0.6881)	unpalatable (0.5385)
	nice (0.6657)	horrible (0.6114)	convienient (0.6279)	incovenient (0.5364)
	really (0.6613)	poor (0.6059)	conveniently (0.5936)	distressing (0.5227)
	this (0.6606)	bad-good (0.6038)	contrived (0.5507)	gore's (0.4977)
	well (0.6427)	b-a-d (0.6023)	convinient (0.5462)	incontrovertable (0.4975)
	interesting (0.6421)	terrible (0.5906)	convienant (0.5443)	embarrassing (0.4888)
	goog (0.6279)	bad.so (0.5898)	coincidental (0.5408)	convenient (0.4880)
	bad (0.6264)	thebad (0.5850)	portable (0.5362)	upsetting (0.4846)
	it (0.6230)	bad-ish (0.5809)	neat (0.5304)	inconvient (0.4784)
SGNS	great (0.7292)	good (0.7190)	Precooked_cocktail_shrimp (0.6684)	convenient (0.5860)
	bad (0.7190)	terrible (0.6829)	easily_accessible (0.6345)	irksome (0.5836)
	terrific (0.6889)	horrible (0.6703)	convenience (0.6247)	annoying (0.5694)
	decent (0.6837)	Bad (0.6699)	user_friendly (0.6226)	problematic (0.5612)
	nice (0.6836)	lousy (0.6648)	Convenient (0.6067)	irritating (0.5519)
	excellent (0.6443)	crummy (0.5678)	inconvenient (0.5860)	unpleasant (0.5427)
	fantastic (0.6408)	horrid (0.5652)	economical (0.5710)	cumbersome (0.5308)
	better (0.6121)	awful (0.5527)	easy (0.5709)	impractical (0.5269)
	solid (0.5806)	dreadful (0.5526)	conveniently_located (0.5673)	costly (0.5226)
	lousy (0.5764)	horrendous (0.5446)	inexpensive (0.5596)	expensive (0.5158)
GV	great (0.8417)	worse (0.7811)	conveniently (0.7695)	troublesome (0.6997)
	better (0.8177)	terrible (0.7739)	convenience (0.7589)	impractical (0.6759)
	very (0.7988)	awful (0.7722)	easy (0.7424)	cumbersome (0.6562)
	nice (0.7975)	horrible (0.7619)	Convenient (0.6922)	problematic (0.6500)
	really (0.7903)	wrong (0.7444)	handy (0.6788)	frustrating (0.6486)

excellent (0.7774)	too (0.7423)	accessible (0.6621)	uncomfortable (0.6338)
decent (0.7734)	worst (0.7384)	hassle-free (0.6485)	bothersome (0.6328)
well (0.7699)	thing (0.7379)	ideal (0.6378)	unavoidable (0.6204)
but (0.7551)	good (0.7355)	economical (0.6251)	inconvenience (0.6178)
much (0.7421)	because (0.7196)	quick (0.6083)	confusing (0.6165)

Table 8.2: Sanity check on domain-specific terms.

	cheap	expensive	quality	price	discount
D1	cheep (0.6788)	pricey (0.7677)	resolution (0.6164)	2.99 (0.6678)	discounts (0.6655)
	inexpensive (0.6067)	pricy (0.6974)	high-quality (0.5984)	4.99 (0.6492)	retailers (0.6277)
	crappy (0.5834)	costly (0.6918)	clarity (0.5934)	1.99 (0.6444)	discounted (0.6161)
	spendy (0.5619)	spendy (0.6906)	presentation (0.5655)	pricetag (0.6380)	14.99 (0.5996)
	amateurish (0.5606)	expencive (0.6642)	performance (0.5579)	1.49 (0.6237)	49.99 (0.5866)
	1.29 (0.5563)	exspensive (0.6614)	all-around (0.5470)	2.50 (0.6131)	200.00 (0.5780)
	1.98 (0.5523)	overpriced (0.6396)	caliber (0.5457)	0.99 (0.6087)	29.95 (0.5719)
	knockoff (0.5384)	over-priced (0.6078)	unwatchable (0.5444)	5.99 (0.6070)	75,000 (0.5705)
	imitation (0.5383)	cost (0.6068)	res (0.5429)	bargain (0.6022)	coupon (0.5675)
	cheesy (0.5347)	scarce (0.5970)	first-rate (0.5424)	99c (0.6002)	sale (0.5669)
D2	cheaply (0.6685)	pricey (0.7469)	workmanship (0.6668)	29.99 (0.7266)	sale (0.7140)
	plastic-y (0.6373)	overpriced (0.6981)	craftsmanship (0.6506)	19.99 (0.7200)	clearance (0.6604)
	inexpensive (0.6161)	pricy (0.6973)	qaulity (0.6398)	39.99 (0.7194)	bargin (0.6532)
	plasticy (0.6090)	costly (0.6762)	quaility (0.6395)	prices (0.7085)	bargain (0.6408)
	cheep (0.6072)	pricier (0.6717)	comfortability (0.6300)	msrp (0.7034)	discounts (0.6294)
	flimsy (0.6051)	exspensive (0.6581)	crystal-clear (0.5865)	pricing (0.6724)	discounted (0.6263)
	cheapy (0.5947)	cheaper (0.6558)	reproduction (0.5797)	-$0.67	9.99 (0.6139)
	cheaper (0.5777)	cost-effective (0.6453)	high-quality (0.5795)	priced (0.6637)	39.99 (0.6044)
	bargin (0.5710)	affordable (0.6280)	clarity (0.5780)	pricepoint (0.6620)	rebate (0.6035)
	chincy (0.5699)	over-priced (0.6260)	g230 (0.5767)	retails (0.6599)	24.99 (0.5983)
D3	cheep (0.8500)	costly (0.8205)	quaility (0.8194)	proce (0.7262)	discounts (0.7553)
	inexpensive (0.7793)	exspensive (0.7252)	quailty (0.7961)	pirce (0.7106)	discounted (0.6752)
	cheapy (0.7032)	pricier (0.7097)	qaulity (0.7743)	pricing (0.6805)	coupon (0.6661)
	chep (0.6776)	expencive (0.7058)	qulaity (0.7715)	pricepoint (0.6736)	sale (0.6175)
	cheaply (0.6770)	expesive (0.6848)	quility (0.7551)	prce (0.6713)	close-out (0.6165)
	flimsy (0.6652)	expenseive (0.6668)	qualiy (0.7390)	value (0.6699)	open-box (0.6135)
	flimsey (0.6540)	pricey (0.6561)	qualty (0.7300)	pric (0.6543)	retailer (0.6117)
	chinsy (0.6434)	cheaper (0.6456)	qualtiy (0.7263)	theprice (0.6410)	promotion (0.6091)
	cheapo (0.6332)	overpriced (0.6450)	quallity (0.7146)	price.it (0.6291)	overstock (0.6056)
	super-cheap (0.6308)	pricy (0.6315)	qality (0.7103)	priceit (0.6248)	costco.com (0.5996)
D4	cheep (0.6863)	costly (0.8188)	quailty (0.7517)	prices (0.7115)	bargain (0.6723)
	trick's (0.6374)	pricey (0.7901)	qulaity (0.7189)	cost (0.6834)	cvs (0.6589)
	cheaply (0.6325)	overpriced (0.7811)	quaility (0.7101)	8.99 (0.6801)	brick-and-mortar (0.6549)
	trick (0.6199)	inexpensive (0.6937)	qaulity (0.7072)	9.99 (0.6723)	bargin (0.6482)
	overpriced (0.5810)	pricy (0.6916)	qualtiy (0.6996)	priced (0.6698)	resale (0.6466)
	second-hand (0.5706)	prohibitively (0.6875)	quiality (0.6593)	2.99 (0.6694)	discounted (0.6378)
	cheapest (0.5623)	over-priced (0.6817)	quallity (0.6586)	19.99 (0.6678)	woolworths (0.6373)
	cheap-o (0.5613)	cheaper (0.6737)	fidelity (0.6480)	13.99 (0.6676)	clearance (0.6338)
	expensive (0.5597)	pricier (0.6715)	qulity (0.6422)	14.99 (0.6671)	walmart (0.6301)
	bargain-bin (0.5545)	spendy (0.6605)	soundquality (0.6304)	7.99 (0.6653)	retail (0.6287)
D5	cheep (0.7472)	costly (0.6849)	quailty (0.8165)	prices (0.7585)	wal-mart (0.7098)
	cheaply (0.7045)	overpriced (0.6726)	qaulity (0.8013)	priced (0.7425)	fye (0.6960)
	inexpensive (0.6787)	affordable (0.6668)	qualtiy (0.7952)	19.99 (0.7149)	walmart (0.6907)
	cheapest (0.6561)	pricey (0.6634)	quaility (0.7776)	9.99 (0.7109)	discounted (0.6900)
	crappy (0.6402)	inexpensive (0.6551)	qualty (0.7702)	12.99 (0.7067)	pre-viewed (0.6737)
	chintzy (0.6311)	pricy (0.6207)	qulaity (0.7522)	49.99 (0.6862)	walgreens (0.6735)
	tacky (0.6303)	prohibitively (0.6191)	qualitythe (0.7304)	theprice (0.6853)	cvs (0.6718)
	bargain-basement (0.6251)	over-priced (0.5952)	quallity (0.7133)	29.99 (0.6812)	big-box (0.6711)
	cheapo (0.6202)	affordably (0.5934)	bluray's (0.7113)	13.99 (0.6801)	bestbuy (0.6673)
	shoddy (0.6199)	limted (0.5904)	quility (0.7088)	5.00 (0.6778)	sale (0.6666)
D6	inexpensive (0.7468)	pricey (0.7737)	quailty (0.6640)	2.99 (0.7882)	discounted (0.7545)
	cheep (0.7240)	costly (0.7111)	quantity (0.6374)	3.99 (0.7755)	discounts (0.6849)
	overpriced (0.6998)	pricy (0.7053)	quaility (0.6353)	0.99 (0.7742)	6.98 (0.6394)
	cheaply (0.6640)	inexpensive (0.6868)	high-quality (0.6307)	99 (0.7595)	costco (0.6343)
	over-priced (0.6604)	cheaper (0.6846)	value (0.6080)	4.99 (0.7435)	sale (0.6299)
	expensive (0.6433)	overpriced (0.6549)	content (0.5951)	1.99 (0.7404)	retail (0.6229)
	pricey (0.6338)	over-priced (0.6443)	quality.the (0.5919)	9.99 (0.7400)	closeout (0.6221)

	cheaper (0.6200)	cheap (0.6433)	qualitiy (0.5880)	0.00 (0.7368)	half-price (0.6171)
	cheapest (0.6106)	affordable (0.6402)	quallity (0.5802)	99c (0.7317)	thrift (0.6164)
	99 (0.6077)	spendy (0.6309)	thequality (0.5791)	5.99 (0.7298)	resell (0.6034)
SGNS	Cheap (0.7455)	pricey (0.7707)	highquality (0.5809)	prices (0.7490)	discounts (0.7701)
	inexpensive (0.7010)	costly (0.7347)	Quality (0.5651)	pricing (0.6457)	discounted (0.7033)
	cheep (0.6507)	pricy (0.6955)	ratios Nonaccruingloans (0.5478)	Prices (0.5965)	Discounts (0.6324)
	cheaper (0.6376)	cheaper (0.6835)	SDVOSB_firms (0.5410)	priced (0.5831)	Discount (0.6239)
	relatively_inexpensive (0.6245)	prohibitively_expensive (0.6804)	quailty (0.5355)	share (0.5386)	Uncle_Mo_bellyache (0.6190)
	reasonably_priced (0.6075)	outrageously_expensive (0.6457)	quaility (0.5211)	premium (0.5309)	coupon (0.5839)
	cheapest (0.5673)	pricier (0.6424)	PowerMax_Looking (0.5003)	priceof (0.5280)	deeply_discounted (0.5642)
	Inexpensive (0.5612)	costlier (0.6265)	reliability (0.5000)	theprice (0.5275)	discounting (0.5342)
	cheaply (0.5575)	expen_sive (0.6254)	centered_neurologic (0.4946)	bellow_resistance (0.5161)	discount_coupons (0.5273)
	bargain_basement (0.5550)	Expensive (0.6216)	ruggedness_reliability (0.4777)	stock (0.5124)	Discounted (0.5264)
GV	cheapest (0.8414)	pricey (0.8702)	high-quality (0.7468)	prices (0.8150)	discounted (0.8414)
	discount (0.7889)	cheaper (0.7994)	exceptional (0.6647)	cost (0.7232)	discounts (0.8058)
	Cheap (0.7698)	costly (0.7889)	Quality (0.6603)	pricing (0.7145)	cheap (0.7889)
	buy (0.7461)	overpriced (0.7372)	high (0.6485)	priced (0.7067)	Discount (0.7599)
	discounted (0.6996)	inexpensive (0.7104)	excellent (0.6399)	buy (0.6702)	cheapest (0.7072)
	inexpensive (0.6813)	afford (0.6953)	top-quality (0.6287)	purchase (0.6536)	purchase (0.6895)
	prices (0.6604)	pricy (0.6935)	superior (0.6251)	lowest (0.6498)	buy (0.6793)
	cheep (0.6569)	cost (0.6820)	best (0.6210)	Price (0.6301)	prices (0.6789)
	Cheapest (0.6435)	high-priced (0.6650)	reliable (0.5899)	cheapest (0.6240)	coupons (0.6444)
	cheaper (0.6286)	priced (0.6614)	superb (0.5870)	discount (0.6240)	coupon (0.6443)

BIBLIOGRAPHY

[1] Margaret Ady, TrustYou, Donna Quadri-Felitti, and Preston Robert Tisch. Consumer research identifies how to present travel review content for more bookings. https://resources.trustyou.com/c/wp-present-travel-review-content?x=0MFT5U, May 2015.

[2] Anish K Agarwal, Vivien Wong, Arthur M Pelullo, Sharath Guntuku, Daniel Polsky, David A Asch, Jonathan Muruako, and Raina M Merchant. Online reviews of specialized drug treatment facilities—identifying potential drivers of high and low patient satisfaction. *Journal of general internal medicine*, pages 1–7, 2019.

[3] Arpita Agnihotri and Saurabh Bhattacharya. Online review helpfulness: Role of qualitative factors. *Psychology & Marketing*, 33(11):1006–1017, 2016.

[4] Alan Agresti and Brent A Coull. Approximate is better than "exact" for interval estimation of binomial proportions. *The American Statistician*, 52(2):119–126, 1998.

[5] Mina Akbarabadi and Monireh Hosseini. Predicting the helpfulness of online customer reviews: The role of title features. *International Journal of Market Research*, page 1470785318819979, December 2018.

[6] Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saied Safaei, Elizabeth D Trippe, Juan B Gutierrez, and Krys Kochut. A brief survey of text mining: Classification, clustering and extraction techniques. *arXiv preprint arXiv:1707.02919*, 2017.

[7] Miriam Alzate, Marta Arce-Urriza, and Javier Cebollada. Exploring the helpfulness of online consumer reviews: The consumer voting journey. *Available at SSRN 3134719*, 2018.

[8] Stefanos Angelidis and Mirella Lapata. Summarizing opinions: Aspect extraction meets sentiment prediction and they are both weakly supervised. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3675–3686, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[9] Nikolay Archak, Anindya Ghose, and Panagiotis G Ipeirotis. Deriving the pricing power of product features by mining consumer reviews. *Management science*, 57(8):1485–1509, 2011.

[10] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. A simple but tough-to-beat baseline for sentence embeddings. In *International Conference on Learning Representations*, 2017.

[11] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. A simple but tough-to-beat baseline for sentence embeddings. *International Conference on Learning Representations*, 2017.

[12] Solomon E Asch. Forming impressions of personality. *The Journal of Abnormal and Social Psychology*, 41(3):258, 1946.

[13] Georgios Askalidis and Edward C. Malthouse. The value of online customer reviews. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys '16, pages 155–158, New York, NY, USA, 2016. ACM.

[14] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of LREC*, volume 10, 01 2010.

[15] H Baek, S Lee, Sehwan Oh, and J.H. Ahn. Normative social influence and online review helpfulness: Polynomial modeling and response surface analysis. *Journal of Electronic Commerce Research*, 16:290–306, 01 2015.

[16] Hyunmi Baek, JoongHo Ahn, and Youngseok Choi. Helpfulness of online consumer reviews: Readers' objectives and review cues. *International Journal of Electronic Commerce*, 17(2):99–126, 2012.

[17] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.

[18] Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. Multimodal machine learning: A survey and taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):423–443, 2018.

[19] Abhijit V Banerjee. A simple model of herd behavior. *The quarterly journal of economics*, 107(3):797–817, 1992.

[20] Shankhadeep Banerjee, Samadrita Bhattacharyya, and Indranil Bose. Whose online reviews to trust? understanding reviewer trustworthiness and its impact on business. *Decision Support Systems*, 96:17–26, 2017.

[21] Mrinal Kanti Baowaly, Yi-Pei Tu, and Kuan-Ta Chen. Predicting the helpfulness

of game reviews: a case study on the steam store. *Journal of Intelligent & Fuzzy Systems*, 36(5):4731–4742, 2019.

[22] Jardeson L.N. Barbosa, Raimundo Santos Moura, and Roney L. de S. Santos. Predicting portuguese steam review helpfulness using artificial neural networks. In *Proceedings of the 22Nd Brazilian Symposium on Multimedia and the Web*, Webmedia '16, pages 287–293, New York, NY, USA, 2016. ACM.

[23] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.

[24] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

[25] Rebekah George Benjamin. Reconstructing readability: Recent developments and recommendations in the analysis of text difficulty. *Educational Psychology Review*, 24(1):63–88, Mar 2012.

[26] Jonah Berger. Does presentation order impact choice after delay? *Topics in Cognitive Science*, 8(3):670–684, 2016.

[27] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*. O'Reilly Media, Inc., 1st edition, 2009.

[28] Einar Bjering, Lars Jaakko Havro, and Øystein Moen. An empirical investigation of self-selection bias and factors influencing review helpfulness. *International Journal of Business and Management*, 10(7):16, 2015.

[29] David Blei, Lawrence Carin, and David Dunson. Probabilistic topic models: A focus on graphical model design and applications to document and image analysis. *IEEE signal processing magazine*, 27(6):55, 2010.

[30] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[31] John Blitzer, Mark Dredze, and Fernando Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 440–447, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[32] Gerlof Bouma. Normalized (pointwise) mutual information in collocation extraction. *Proceedings of GSCL*, pages 31–40, 2009.

[33] Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. Concreteness ratings for 40 thousand generally known english word lemmas. *Behavior Research Methods*, 46(3):904–911, oct 2013.

[34] BuiltWith. Distribution for websites using e-commerce technologies. `https://trends.builtwith.com/shop/traffic/Entire-Internet`, January 2020.

[35] Jose Camacho-Collados, Claudio Delli Bovi, Luis Espinosa-Anke, Sergio Oramas, Tommaso Pasini, Enrico Santus, Vered Shwartz, Roberto Navigli, and Horacio Saggion. SemEval-2018 task 9: Hypernym discovery. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 712–724, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[36] Jose Camacho-Collados and Mohammad Taher Pilehvar. From word to sense embeddings: A survey on vector representations of meaning. *Journal of Artificial Intelligence Research*, 63:743–788, 2018.

[37] Erik Cambria, Daniel Olsher, and Dheeraj Rajagopal. Senticnet 3: a common and common-sense knowledge base for cognition-driven sentiment analysis. In *Twenty-eighth AAAI conference on artificial intelligence*, 2014.

[38] Qing Cao, Wenjing Duan, and Qiwei Gan. Exploring determinants of voting for the "helpfulness" of online user reviews: A text mining approach. *Decision Support Systems*, 50(2):511–521, 2011.

[39] Pew Research Center. Online shopping and e-commerce. `http://www.pewinternet.org/2016/12/19/online-shopping-and-e-commerce/`, 2016.

[40] Jeanne Sternlicht Chall and Edgar Dale. *Readability revisited: The new Dale-Chall readability formula*. Brookline Books, 1995.

[41] E. B. Charrada. Which one to read? factors influencing the usefulness of online reviews for re. In *2016 IEEE 24th International Requirements Engineering Conference Workshops (REW)*, pages 46–52, Sep. 2016.

[42] Swagato Chatterjee. Drivers of helpfulness of online hotel reviews: A sentiment and emotion mining approach. *International Journal of Hospitality Management*, page 102356, 2019.

[43] Cen Chen, Minghui Qiu, Yinfei Yang, Jun Zhou, Jun Huang, Xiaolong Li, and Forrest Sheng Bao. Multi-domain gated cnn for review helpfulness prediction. In *The World Wide Web Conference*, WWW '19, pages 2630–2636, New York, NY, USA, 2019. ACM.

[44] Cen Chen, Yinfei Yang, Jun Zhou, Xiaolong Li, and Forrest Sheng Bao. Cross-domain review helpfulness prediction based on convolutional neural networks with auxiliary domain discriminators. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 602–607, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[45] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. Neural attentional rating regression with review-level explanations. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, pages 1583–1592, Republic and Canton of Geneva, Switzerland, 2018. International World Wide Web Conferences Steering Committee.

[46] Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. A survey on dialogue systems: Recent advances and new frontiers. *Acm Sigkdd Explorations Newsletter*, 19(2):25–35, 2017.

[47] Jie Chen, Chunxia Zhang, and Zhendong Niu. Identifying helpful online reviews with word embedding features. In Franz Lehner and Nora Fteimi, editors, *Knowledge Science, Engineering and Management*, pages 123–133, Cham, 2016. Springer International Publishing.

[48] Yi-Hsiu Cheng and Hui-Yi Ho. Social influence's impact on reader perceptions of online reviews. *Journal of Business Research*, 68(4):883–887, 2015. Special Issue on Global entrepreneurship and innovation in management.

[49] Judith A. Chevalier and Dina Mayzlin. The effect of word of mouth on sales: Online book reviews. *Journal of Marketing Research*, 43(3):345–354, 2006.

[50] Giulia Chiriatti, Dominique Brunato, Felice Dell'Orletta, and Giulia Venturi. What makes a review helpful? predicting the helpfulness of italian tripadvisor reviews. In Raffaella Bernardi, Roberto Navigli, and Giovanni Semeraro, editors, *Italian Conference on Computational Linguistics (CLiC-it)*, Bari, Italy, November 2019. CEUR.

[51] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase repre-

sentations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[52] Sungwoo Choi and Anna S. Mattila. The effect of experience congruity on repurchase intention: The moderating role of public commitment. *Service Science*, 10(2):124–138, 2018.

[53] François Chollet et al. Keras. `https://github.com/fchollet/keras`, 2015.

[54] Alton Y.K. Chua and Snehasish Banerjee. Helpfulness of user-generated reviews as a function of review sentiment, product type and information quality. *Computers in Human Behavior*, 54:547–554, 2016.

[55] Robert B Cialdini and Noah J Goldstein. Social influence: Compliance and conformity. *Annu. Rev. Psychol.*, 55:591–621, 2004.

[56] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

[57] LH Coenen, Liselotte Hedebouw, and Gün R Semin. The linguistic category model (lcm). *Manual. Free University Amsterdam*, pages 1151434261594–8567, 2006.

[58] Meri Coleman and Ta Lin Liau. A computer readability formula designed for machine scoring. *Journal of Applied Psychology*, 60(2):283, 1975.

[59] Kevyn Collins-Thompson. Computational assessment of text readability: A survey of current and future research. *ITL-International Journal of Applied Linguistics*, 165(2):97–135, 2014.

[60] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537, 2011.

[61] L. Connors, S. M. Mudambi, and D. Schuff. Is it the review or the reviewer? a multi-method approach to determine the antecedents of online review helpfulness. In *2011 44th Hawaii International Conference on System Sciences*, pages 1–10, Jan 2011.

[62] KPMG International Cooperative. Global online consumer re-

port. https://assets.kpmg/content/dam/kpmg/xx/pdf/2017/01/the-truth-about-online-consumers.pdf, January 2017.

[63] Dan Cosley, Shyong K. Lam, Istvan Albert, Joseph A. Konstan, and John Riedl. Is seeing believing?: How recommender system interfaces affect users' opinions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, pages 585–592, New York, NY, USA, 2003. ACM.

[64] Crowdtap and Ipsos. Social influence: Marketing's new frontier. https://ivetriedthat.com/wp-content/uploads/2015/02/Social_Influence_Research_Paper.pdf, March 2014.

[65] Jan Jacob Cuilenburg, Jan Kleinnijenhuis, and Johannes Abraham Ridder. *Tekst en betoog: naar een computer gestuurde inhoudsanalyse van betogende teksten*. Coutinho, 1988.

[66] Andrew M Dai, Christopher Olah, and Quoc V Le. Document embedding with paragraph vectors. *arXiv preprint arXiv:1507.07998*, 2015.

[67] Jurafsky Dan and Martin James H. Speech and language processing (third edition draft). https://web.stanford.edu/~jurafsky/slp3/.

[68] Cristian Danescu-Niculescu-Mizil, Gueorgi Kossinets, Jon Kleinberg, and Lillian Lee. How opinions are received by online communities: A case study on amazon.com helpfulness votes. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pages 141–150, New York, NY, USA, 2009. ACM.

[69] Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 933–941. JMLR. org, 2017.

[70] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990.

[71] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[72] Debasmita Dey and Pradeep Kumar. A novel approach to identify the determi-

nants of online review helpfulness and predict the helpfulness score across product categories. In Sanjay Madria, Philippe Fournier-Viger, Sanjay Chaudhary, and P. Krishna Reddy, editors, *Big Data Analytics*, pages 365–388, Cham, 2019. Springer International Publishing.

[73] Ruihai Dong, Markus Schaal, and Barry Smyth. Topic extraction from online reviews for classification and recommendation. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI 13)*, pages 1310–1316, June 2013.

[74] Jiahua Du, Sandra Michalska, Sudha Subramani, Hua Wang, and Yanchun Zhang. Neural attention with character embeddings for hay fever detection from twitter. *Health Information Science and Systems*, 7(1):21, October 2019.

[75] Jiahua Du, Jia Rong, Sandra Michalska, Hua Wang, and Yanchun Zhang. Feature selection for helpfulness prediction of online product reviews: An empirical study. *PLOS ONE*, 14(12):1–26, December 2019.

[76] Jiahua Du, Jia Rong, Hua Wang, and Yanchun Zhang. Exploiting review neighbors for contextualized helpfulness prediction. Submitted to Decision Support Systems.

[77] Jiahua Du, Jia Rong, Hua Wang, and Yanchun Zhang. Helpfulness prediction for online reviews with explicit content-rating interaction. In *Web Information Systems Engineering (WISE)*, pages 795–809, Hong Kong SAR, China, October 2019. Springer International Publishing.

[78] Jiahua Du, Liping Zheng, Jiantao He, Jia Rong, Hua Wang, and Yanchun Zhang. An interactive network for end-to-end review helpfulness modeling. *Data Science and Engineering*, pages 1–19, 2020.

[79] William H DuBay. *Smart Language: Readers, Readability, and the Grading of Text.* ERIC, 2007.

[80] Bjering Einar. Online consumer reviews: The moderating effect of product category. Master's thesis, Norwegian University of Science and Technology, June 2014.

[81] eMarketer. Moms place trust in other consumers. `https://www.emarketer.com/Article/Moms-Place-Trust-Other-Consumers/1007509`, February 2010.

[82] Episerver. Online shopping habits and retailer strategies. https://www.episerver.com/reports/reimagining-commerce-report/, December 2019.

[83] Enes Eryarsoy and Selwyn Piramuthu. Experimental evaluation of sequential bias in online customer reviews. *Information & Management*, 51(8):964–971, 2014.

[84] Seyed Pouyan Eslami, Maryam Ghasemaghaei, and Khaled Hassanein. Which online reviews do consumers find most helpful? a multi-method investigation. *Decision Support Systems*, 113:32–42, 2018.

[85] M. Fan, Y. Feng, M. Sun, P. Li, H. Wang, and J. Wang. Multi-task neural learning architecture for end-to-end identification of helpful reviews. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 343–350, Aug 2018.

[86] Miao Fan, Chao Feng, Lin Guo, Mingming Sun, and Ping Li. Product-aware helpfulness prediction of online reviews. In *The World Wide Web Conference*, WWW '19, pages 2715–2721, New York, NY, USA, 2019. ACM, ACM.

[87] Bin Fang, Qiang Ye, Deniz Kucukusta, and Rob Law. Analysis of the perceived value of online tourism reviews: Influence of readability and reviewer characteristics. *Tourism Management*, 52:498–506, 2016.

[88] James N Farr, James J Jenkins, and Donald G Paterson. Simplification of flesch reading ease formula. *Journal of applied psychology*, 35(5):333, 1951.

[89] February. Digital investments pay off for walmart in e-commerce race. https://www.emarketer.com/content/digital-investments-pay-off-for-walmart-in-ecommerce-race, eMarketer 2019.

[90] Christiane Fellbaum. *WordNet: An electronic lexical database (Language, Speech, and Communication).* The MIT Press, 05 1998.

[91] Raffaele Filieri. What makes an online consumer review trustworthy? *Annals of Tourism Research*, 58:46–64, 2016.

[92] J. Firth. A synopsis of linguistic theory 1930-1955. In *Studies in Linguistic Analysis*. Philological Society, Oxford, 1957.

[93] Jorge E. Fresneda and David Gefen. A semantic measure of online review

helpfulness and the importance of message entropy. *Decision Support Systems*, 125:113117, 2019.

[94] Jorge E. Fresneda and David Gefen. A semantic measure of online review helpfulness and the importance of message entropy. *Decision Support Systems*, 125:113117, 2019.

[95] Fan & Fuel. No online customer reviews means big problems. https://fanandfuel.com/no-online-customer-reviews-means-big-problems-2017/, December 2016.

[96] Suyu Ge, Tao Qi, Chuhan Wu, Fangzhao Wu, Xing Xie, and Yongfeng Huang. Helpfulness-aware review based neural recommendation. *CCF Transactions on Pervasive Computing and Interaction*, Nov 2019.

[97] A. Ghose and P. G. Ipeirotis. Estimating the helpfulness and economic impact of product reviews: Mining text and reviewer characteristics. *IEEE Transactions on Knowledge and Data Engineering*, 23(10):1498–1512, Oct 2011.

[98] Anindya Ghose and Panagiotis G. Ipeirotis. Designing novel review ranking systems: Predicting the usefulness and impact of reviews. In *Proceedings of the Ninth International Conference on Electronic Commerce*, ICEC '07, page 303–310, New York, NY, USA, 2007. Association for Computing Machinery.

[99] Leilani H Gilpin, David Bau, Ben Z Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*, pages 80–89. IEEE, 2018.

[100] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

[101] David Godes and José C. Silva. Sequential and temporal dynamics of online opinion. *Marketing Science*, 31(3):448–473, 2012.

[102] Przemyslaw A Grabowicz, Francisco Romero-Ferrero, Theo Lins, Gonzalo G de Polavieja, Fabrício Benevenuto, and Krishna P Gummadi. An experimental study of opinion influenceability. *arXiv preprint arXiv:1512.00770*, 2015.

[103] Robert Gunning. The fog index after twenty years. *Journal of Business Communication*, 6(2):3–13, 1969.

[104] Bin Guo and Shasha Zhou. Understanding the impact of prior reviews on subsequent reviews: The role of rating volume, variance and reviewer characteristics. *Electronic Commerce Research and Applications*, 20:147–158, 2016.

[105] Daomeng Guo, Yang Zhao, Liyi Zhang, Xuan Wen, and Cong Yin. Conformity feedback in an online review helpfulness evaluation task leads to less negative feedback-related negativity amplitudes and more positive p300 amplitudes. *Journal of Neuroscience, Psychology, and Economics*, 12(2), 2019.

[106] Mark Andrew Hall. *Correlation-based feature selection for machine learning*. PhD thesis, University of Waikato Hamilton, 1999.

[107] Hamad, Musab Umair Malik, and Khalid Iqbal. Review helpfulness as a function of linguistic indicators. In *International Journal of Computer Science and Network Security (IJCSNS)*, 2018.

[108] Mark A Hamilton and Kristine L Nowak. Information systems concepts across two decades: An empirical analysis of trends in theory, methods, process, and research domains. *Journal of Communication*, 55(3):529–553, 2005.

[109] Robin Hanna. Amazon on a positive note: The end of downvoting. `https://sellics.com/blog-amazon-on-a-positive-note-the-end-of-downvoting/`, January 2020.

[110] Md. Enamul Haque, Mehmet Engin Tozal, and Aminul Islam. Helpfulness prediction of online product reviews. In *Proceedings of the ACM Symposium on Document Engineering 2018*, DocEng '18, pages 35:1–35:4, New York, NY, USA, 2018. ACM.

[111] Zellig S Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.

[112] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[113] Ruining He and Julian McAuley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, pages 507–517,

Republic and Canton of Geneva, Switzerland, 2016. International World Wide Web Conferences Steering Committee.

[114] Diana C Hernandez-Bocanegra and Jürgen Ziegler. Assessing the helpfulness of review content for explaining recommendations. In *EARS Workshop at SIGIR'19*, 2019.

[115] Ben Hixon, Peter Clark, and Hannaneh Hajishirzi. Learning knowledge graphs for question answering through conversational dialog. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 851–861, Denver, Colorado, May–June 2015. Association for Computational Linguistics.

[116] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[117] Anne-Sophie Hoffait, Ashwin Ittoo, and Michael Schyns. Assessing and predicting review helpfulness: Critical review, open challenges and research agenda. In *29ème conférence européenne sur la recherche opérationnelle (EURO2018)*, 2018.

[118] Katie Hollar. From reviews to results: The impact of business software reviews. `https://www.capterra.com/b2b-software-reviews-infographic`, May 2015.

[119] Hong Hong, Di Xu, G. Alan Wang, and Weiguo Fan. Understanding the determinants of online review helpfulness: A meta-analytic investigation. *Decision Support Systems*, 102:1–11, 2017.

[120] Yu Hong, Jun Lu, Jianmin Yao, Qiaoming Zhu, and Guodong Zhou. What reviews are satisfactory: Novel features for automatic helpfulness voting. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, pages 495–504, New York, NY, USA, 2012. ACM.

[121] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM, 2004.

[122] Ya-Han Hu and Kuanchin Chen. Predicting hotel review helpfulness: The impact of review visibility, and interaction between hotel stars and review ratings. *International Journal of Information Management*, 36(6, Part A):929–944, 2016.

[123] Ya-Han Hu, Kuanchin Chen, and Pei-Ju Lee. The effect of user-controllable filters on the prediction of online hotel reviews. *Information & Management*, 54(6):728–744, 2017. Smart Tourism: Traveler, Business, and Organizational Perspectives.

[124] Albert H. Huang, Kuanchin Chen, David C. Yen, and Trang P. Tran. A study of factors that contribute to online review helpfulness. *Computers in Human Behavior*, 48:17–27, 2015.

[125] Albert H. Huang and David C. Yen. Predicting the helpfulness of online reviews—a replication. *International Journal of Human–Computer Interaction*, 29(2):129–138, 2013.

[126] Clayton J Hutto and Eric Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *International AAAI Conference on Web and Social Media*, pages 216–225, May 2014.

[127] San-Yih Hwang, Chia-Yu Lai, Jia-Jhe Jiang, and Shanlin Chang. The identification of noteworthy hotel reviews for hotel management. In *Pacific Asia Journal of the Association for Information Systems (PACIS)*, pages 1–17, December 2014.

[128] Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1681–1691, Beijing, China, July 2015. Association for Computational Linguistics.

[129] Stone Philip J., Bales Robert F., Namenwirth J. Zvi, and Ogilvie Daniel M. The general inquirer: A computer system for content analysis and retrieval based on the sentence as a unit of information. *Behavioral Science*, 7(4):484–498, 1966.

[130] Nitin Jindal and Bing Liu. Mining comparative sentences and relations. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 2*, AAAI'06, pages 1331–1336. AAAI Press, 2006.

[131] Nitin Jindal and Bing Liu. Opinion spam and analysis. In *Proceedings of the 2008 international conference on web search and data mining*, pages 219–230. ACM, 2008.

[132] Li Jing, Xu Xin, and Eric Ngai. An examination of the joint impacts of review content and reviewer characteristics on review usefulness-the case of yelp. com. In *22nd Americas Conference on Information Systems (AMCIS)*, pages 1–5, 2016.

[133] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, Valencia, Spain, April 2017. Association for Computational Linguistics.

[134] Rafal Jozefowicz, Wojciech Zaremba, and Ilya Sutskever. An empirical exploration of recurrent network architectures. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2342–2350, Lille, France, 07–09 Jul 2015. PMLR.

[135] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 655–665, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

[136] Gaurav Kapoor and Selwyn Piramuthu. Sequential bias in online product reviews. *Journal of Organizational Computing and Electronic Commerce*, 19(2):85–95, 2009.

[137] Sahar Karimi and Fang Wang. Online review helpfulness: Impact of reviewer profile image. *Decision Support Systems*, 96:39–48, 2017.

[138] Rachita Kashyap and Abhilash Ponnam. Conceptualising a formative model for online review helpfulness: Proposal. *The Marketing Review*, 19(1-2):107–125, November 2019.

[139] Rimma Kats. Surprise! most consumers look at reviews before a purchase. `https://www.emarketer.com/content/surprise-most-consumers-look-at-reviews-before-a-purchase`, Feburary 2018.

[140] Foo Sheng Khoo, Phoey Lee Teh, and Pei Boon Ooi. Consistency of online consumers' perceptions of posted comments: An analysis of tripadvisor of reviews. *Journal of information and Communication Technology*, 4:374–393, 07 2017.

[141] Soo-Min Kim, Patrick Pantel, Tim Chklovski, and Marco Pennacchiotti. Automatically assessing review helpfulness. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP '06, pages 423–430, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

[142] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, October 2014. Association for Computational Linguistics.

[143] Yoon Kim, Yacine Jernite, David Sontag, and Alexander M. Rush. Character-aware neural language models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pages 2741–2749. AAAI Press, 2016.

[144] J. P. Kincaid, J. A. Aagard, J. W. O'Hara, and L. K. Cottrell. Computer readability editing system. *IEEE Transactions on Professional Communication*, PC-24(1):38–42, March 1981.

[145] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[146] Prasadith Kirinde Gamaarachchige and Diana Inkpen. Multi-task, multi-channel, multi-input learning for mental illness detection using social media text. In *Proceedings of the Tenth International Workshop on Health Text Mining and Information Analysis (LOUHI 2019)*, pages 54–64, Hong Kong, November 2019. Association for Computational Linguistics.

[147] Nicholas Kitonyi. Uk online shopping and e-commerce statistics. `https://www.nasdaq.com/articles/uk-online-shopping-and-e-commerce-statistics-2017-2017-03-14`, March 2017.

[148] Nikolaos Korfiatis, Elena García-Bariocanal, and Salvador Sánchez-Alonso. Evaluating content quality and helpfulness of online product reviews: The interplay of review helpfulness vs. review content. *Electronic Commerce Research and Applications*, 11(3):205–217, 2012.

[149] Jiaying Kou, Xiaoming Fu, Jiahua Du, Hua Wang, and Geordie Z Zhang. Understanding housing market behaviour from a microscopic perspective. In *2018 27th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–9. IEEE, July 2018.

[150] Stavroula-Thaleia Kousta, Gabriella Vigliocco, David P Vinson, Mark Andrews, and Elena Del Campo. The representation of abstract words: why emotion matters. *Journal of Experimental Psychology: General*, 140(1):14, 2011.

[151] Robert V. Kozinets. Amazonian Forests and Trees: Multiplicity and Objectivity

in Studies of Online Consumer-Generated Ratings and Reviews, A Commentary on de Langhe, Fernbach, and Lichtenstein. *Journal of Consumer Research*, 42(6):834–839, 04 2016.

[152] Srikumar Krishnamoorthy. Linguistic features for review helpfulness prediction. *Expert Systems with Applications*, 42(7):3751–3759, 2015.

[153] Kevin Kuan, Kai-Lung Hui, Pattarawan Prasarnphanich, and Hok-Yin Lai. What makes a review voted? an empirical investigation of review voting in online review systems. *Journal of the Association for Information Systems*, 16:48–71, 01 2015.

[154] Miron B. Kursa and Witold R. Rudnicki. Feature Selection with the Boruta Package. *Journal of Statistical Software*, 36(i11):1–13, September 2010.

[155] Linchi Kwok and Karen L. Xie. Factors contributing to the helpfulness of online hotel reviews: Does manager response play a role? *International Journal of Contemporary Hospitality Management*, 28(10):2156–2177, 2016.

[156] Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. Recurrent convolutional neural networks for text classification. In *Twenty-ninth AAAI conference on artificial intelligence*, 2015.

[157] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196, 2014.

[158] Minwoo Lee, Miyoung Jeong, and Jongseo Lee. Roles of negative emotions in customers' perceived helpfulness of hotel reviews on a user-generated review website: A text mining approach. *International Journal of Contemporary Hospitality Management*, 29(2):762–783, 2017.

[159] Moontae Lee, Seok Hyun Jin, and David Mimno. Beyond exchangeability: The chinese voting process. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 4934–4942. Curran Associates, Inc., 2016.

[160] Pei-Ju Lee, Ya-Han Hu, and Kuan-Ting Lu. Assessing the helpfulness of online hotel reviews: A classification-based approach. *Telematics and Informatics*, 35(2):436–445, 2018.

[161] Sangjae Lee and Joon Yeon Choeh. Predicting the helpfulness of online reviews

using multilayer perceptron neural networks. *Expert Systems with Applications*, 41(6):3041–3046, 2014.

[162] Sangjae Lee and Joon Yeon Choeh. Exploring the determinants of and predicting the helpfulness of online user reviews using decision trees. *Management Decision*, 55(4):681–700, 2017.

[163] Tao Lei, Regina Barzilay, and Tommi Jaakkola. Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 107–117, Austin, Texas, November 2016. Association for Computational Linguistics.

[164] Jiwei Li. Hotel review datasets. `http://www.cs.cmu.edu/~jiweil/html/hotel-review.html`, May 2014.

[165] Mengxiang Li, Liqiang Huang, Chuan-Hoo Tan, and Kwok-Kee Wei. Helpfulness of online product reviews as seen by consumers: Source and content features. *International Journal of Electronic Commerce*, 17(4):101–136, 2013.

[166] Sheng-Tun Li, Thuong-Thi Pham, and Hui-Chi Chuang. Do reviewers' words affect predicting their helpfulness ratings? locating helpful reviewers by linguistics styles. *Information & Management*, 56(1):28–38, 2019.

[167] Michele Linn. How content influences the purchasing process. `https://contentmarketinginstitute.com/2017/07/content-influences-purchasing-research/`, July 2017.

[168] Marco Lippi and Paolo Torroni. Margot: A web server for argumentation mining. *Expert System Application*, 65:292–303, 2016.

[169] Andrew Lipsman. Global ecommerce. `https://www.emarketer.com/content/global-ecommerce-2019`, June 2019.

[170] Zachary C Lipton, John Berkowitz, and Charles Elkan. A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*, 2015.

[171] Haijing Liu, Yang Gao, Pin Lv, Mengxue Li, Shiqiang Geng, Minglan Li, and Hao Wang. Using argument-based features to predict and analyse review helpfulness. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1358–1363, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[172] Jingjing Liu, Yunbo Cao, Chin-Yew Lin, Yalou Huang, and Ming Zhou. 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)*, pages 334–342, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[173] Qianqian Liu and Elena Karahanna. An agent-based modeling analysis of helpful vote on online product reviews. In *48th Hawaii International Conference on System Sciences*, pages 1585–1595, Jan 2015.

[174] Xiaohua Liu, Shaodian Zhang, Furu Wei, and Ming Zhou. Recognizing named entities in tweets. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 359–367, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[175] Y. Liu, X. Huang, A. An, and X. Yu. Modeling and predicting the helpfulness of online reviews. In *2008 Eighth IEEE International Conference on Data Mining*, pages 443–452, Dec 2008.

[176] Yang Liu, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. Topical word embeddings. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[177] Yao Liu, Cuiqing Jiang, Yong Ding, Zhao Wang, Xiaozhong Lv, and Junhua Wang. Identifying helpful quality-related reviews from social media based on attractive quality theory. *Total Quality Management & Business Excellence*, 0(0):1–20, 2017.

[178] Yiming Liu, Xuezhi Cao, and Yong Yu. Are you influenced by others when rating?: Improve rating prediction by conformity modeling. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys '16, pages 269–272, New York, NY, USA, 2016. ACM.

[179] Ying Liu, Jian Jin, Ping Ji, Jenny A. Harding, and Richard Y.K. Fung. Identifying helpful online reviews: A product designer's perspective. *Computer-Aided Design*, 45(2):180–194, 2013.

[180] Zhiwei Liu and Sangwon Park. What makes a useful online review? implication for travel product websites. *Tourism Management*, 47:140–151, 2015.

[181] Inés López-López and José Francisco Parra. Is a most helpful ewom review really helpful? the impact of conflicting aggregate valence and consumer's goals on product attitude. *Internet Research*, 26(4):827–844, 2016.

[182] Yue Lu, Panayiotis Tsaparas, Alexandros Ntoulas, and Livia Polanyi. Exploiting social context for review quality prediction. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 691–700, New York, NY, USA, 2010. ACM.

[183] Yi Luo and Xiaowei Xu. Predicting the Helpfulness of Online Restaurant Reviews Using Different Machine Learning Algorithms: A Case Study of Yelp. *Sustainability*, 11(19):1–17, September 2019.

[184] Bernhard Lutz, Nicolas Pröllochs, and Dirk Neumann. Understanding the role of two-sided argumentation in online consumer reviews: A language-based perspective. In *Thirty ninth International Conference on Information Systems*, 2018.

[185] Hao Ma, Dengyong Zhou, Chao Liu, Michael R Lyu, and Irwin King. Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 287–296. ACM, 2011.

[186] Hao Ma, Dengyong Zhou, Chao Liu, Michael R. Lyu, and Irwin King. Recommender systems with social regularization. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, WSDM '11, page 287–296, New York, NY, USA, 2011. Association for Computing Machinery.

[187] Yufeng Ma, Zheng Xiang, Qianzhou Du, and Weiguo Fan. Effects of user-provided photos on hotel review helpfulness: An analytical approach with deep leaning. *International Journal of Hospitality Management*, 71:120–131, 2018.

[188] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[189] M.S.I. Malik and Ayyaz Hussain. Helpfulness of product reviews as a function of discrete positive and negative emotions. *Computers in Human Behavior*, 73:290–302, 2017.

[190] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[191] Luciana B. Maroun, Mirella M. Moro, Jussara M. Almeida, and Ana Paula C. Silva. Assessing review recommendation techniques under a ranking perspective. In *Proceedings of the 27th ACM Conference on Hypertext and Social Media*, HT '16, pages 113–123, New York, NY, USA, 2016. ACM.

[192] Lionel Martin and Pearl Pu. Prediction of helpful reviews using emotions extraction. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI'14, pages 1551–1557. AAAI Press, 2014.

[193] Paolo Massa and Paolo Avesani. Trust-aware recommender systems. In *Proceedings of the 2007 ACM conference on Recommender systems*, pages 17–24, 2007.

[194] G Harry Mc Laughlin. Smog grading–a new readability formula. *Journal of reading*, 12(8):639–646, 1969.

[195] Julian McAuley and Jure Leskovec. Hidden factors and hidden topics: Understanding rating dimensions with review text. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, pages 165–172, New York, NY, USA, 2013. ACM.

[196] Santiago Melián-González, Jacques Bulchand-Gidumal, and Beatriz González López-Valcárcel. Online customer reviews of hotels: As participation increases, better evaluation is obtained. *Cornell Hospitality Quarterly*, 54(3):274–283, 2013.

[197] Yuan Meng, Hongwei Wang, and Lijuan Zheng. Impact of online word-of-mouth on sales: the moderating role of product review quality. *New Review of Hypermedia and Multimedia*, 24(1):1–27, 2018.

[198] Hugo Mercier and Olivier Morin. Majority rules: how good are we at aggregating convergent opinions? *Evolutionary Human Sciences*, 1, 2019.

[199] Matthias Mertz, Nikolaos Korfiatis, and Roberto V. Zicari. Using dependency bigrams and discourse connectives for predicting the helpfulness of online reviews. In Martin Hepp and Yigal Hoffner, editors, *E-Commerce and Web Technologies*, pages 146–152, Cham, 2014. Springer International Publishing.

[200] Tomáš Mikolov, Jeff Dean, Quoc Le, Thomas Strohmann, and Claudio Baecchi. Learning representations of text using neural networks. In *NIPS Deep Learning Workshop*, pages 1–31, 2013.

[201] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[202] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in

continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia, June 2013. Association for Computational Linguistics.

[203] WENDY W. MOE and MICHAEL TRUSOV. The value of social dynamics in online product ratings forums. *Journal of Marketing Research*, 48(3):444–456, 2011.

[204] Samaneh Moghaddam, Mohsen Jamali, and Martin Ester. Etf: Extended tensor factorization model for personalizing prediction of review helpfulness. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, WSDM '12, pages 163–172, New York, NY, USA, 2012. ACM.

[205] Saif M Mohammad and Peter D Turney. Crowdsourcing a word–emotion association lexicon. *Computational Intelligence*, 29(3):436–465, 2013.

[206] Elaheh Momeni, Claire Cardie, and Nicholas Diakopoulos. A survey on assessment and ranking methodologies for user-generated content on the web. *ACM Comput. Surv.*, 48(3):41:1–41:49, December 2015.

[207] Elaheh Momeni, Claire Cardie, and Nicholas Diakopoulos. How to assess and rank user-generated content on web. In *Companion Proceedings of the The Web Conference 2018*, WWW '18, pages 489–493, Republic and Canton of Geneva, Switzerland, 2018. International World Wide Web Conferences Steering Committee.

[208] Sarah G. Moore and Katherine C. Lafreniere. How online word-of-mouth impacts receivers. *Consumer Psychology Review*, 3(1):34–59, 2020.

[209] Mohammadreza Mousavizadeh, Mehrdad Koohikamali, and Mohammad Salehan. The effect of central and peripheral cues on online review helpfulness: A comparison between functional and expressive products. In *Thirty Sixth International Conference on Information Systems, Fort Worth (ICIS)*, pages 1–22, 2015.

[210] Jiaqi Mu and Pramod Viswanath. All-but-the-top: Simple and effective post-processing for word representations. In *International Conference on Learning Representations*, 2018.

[211] Lev Muchnik, Sinan Aral, and Sean J. Taylor. Social influence bias: A randomized experiment. *Science*, 341(6146):647–651, 2013.

[212] Susan M. Mudambi and David Schuff. Research note: What makes a helpful online review? a study of customer reviews on amazon.com. *MIS Quarterly*, 34(1):185–200, 2010.

[213] Subhabrata Mukherjee, Kashyap Popat, and Gerhard Weikum. Exploring latent semantic factors to find useful product reviews. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pages 480–488, 2017.

[214] Rosie Murphy. Local consumer review survey. https://www.brightlocal.com/learn/local-consumer-review-survey-2016/, 2016.

[215] Rosie Murphy. Local consumer review survey. https://www.brightlocal.com/learn/local-consumer-review-survey-2017/, 2017.

[216] Rosie Murphy. Local consumer review survey. https://www.brightlocal.com/learn/local-consumer-review-survey-2018/, 2018.

[217] Rosie Murphy. Local consumer review survey. https://www.brightlocal.com/research/local-consumer-review-survey/, 2018.

[218] Ashwini Murthy. How many e-commerce companies are there? what's the global e-commerce market size? https://blog.pipecandy.com/e-commerce-companies-market-size/?utm_source=ecom_prod_cat_blog&utm_medium=site_link&utm_campaign=blog, October 2017.

[219] Makoto Nakayama and Yun Wan. An exploratory study:" blind-testing" consumers how they rate helpfulness of online reviews. In *International Conference on Information Resources Management*, 2012.

[220] Thomas L. Ngo-Ye and Atish P. Sinha. The influence of reviewer engagement characteristics on online review helpfulness: A text regression model. *Decision Support Systems*, 61:47–58, 2014.

[221] Thomas L. Ngo-Ye, Atish P. Sinha, and Arun Sen. Predicting the helpfulness of online reviews using a scripts-enriched text regression model. *Expert Systems with Applications*, 71:98–110, 2017.

[222] Jianmo Ni, Jiacheng Li, and Julian McAuley. 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)*, pages 188–197, 2019.

[223] Finn Årup Nielsen. A new ANEW: evaluation of a word list for sentiment analysis in microblogs. *CoRR*, abs/1103.2903, 2011.

[224] Gerardo Ocampo Diaz and Vincent Ng. 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)*, pages 698–708, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[225] Michael P. O'Mahony and Barry Smyth. A classification-based review recommender. In Max Bramer, Richard Ellis, and Miltos Petridis, editors, *Research and Development in Intelligent Systems XXVI*, pages 49–62, London, 2010. Springer London.

[226] Jahna Otterbacher. 'helpfulness' in online communities: A measure of message quality. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 955–964, New York, NY, USA, 2009. ACM.

[227] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[228] Lionel Page and Katie Page. Last shall be first: A field study of biases in sequential performance evaluation on the idol series. *Journal of Economic Behavior & Organization*, 73(2):186–198, 2010.

[229] Yue Pan and Jason Q. Zhang. Born unequal: A study of the helpfulness of user-generated product reviews. *Journal of Retailing*, 87(4):598–612, 2011.

[230] Sangwon Park and Juan L. Nicolau. Asymmetric effects of online consumer reviews. *Annals of Tourism Research*, 50:67–83, 2015.

[231] Yoon-Joo Park. Predicting the helpfulness of online customer reviews across different product types. *Sustainability*, 10(6), 2018.

[232] Marco Passon, Marco Lippi, Giuseppe Serra, and Carlo Tasso. Predicting the usefulness of Amazon reviews using off-the-shelf argumentation mining. In *Proceedings of the 5th Workshop on Argument Mining*, pages 35–39, Brussels, Belgium, November 2018. Association for Computational Linguistics.

[233] Debanjan Paul, Sudeshna Sarkar, Muthusamy Chelliah, Chetan Kalyan, and Prajit Prashant Sinai Nadkarni. Recommendation of high quality representative re-

views in e-commerce. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, RecSys '17, pages 311–315, New York, NY, USA, 2017. ACM.

[234] Heiko Paulheim. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic web*, 8(3):489–508, 2017.

[235] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[236] James W Pennebaker, Ryan L Boyd, Kayla Jordan, and Kate Blackburn. The development and psychometric properties of liwc2015. Technical report, The University of Texas at Austin, 2015.

[237] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[238] Iryna Pentina, Ainsworth Anthony Bailey, and Lixuan Zhang. Exploring effects of source similarity, message valence, and receiver regulatory focus on yelp review persuasiveness and purchase intentions. *Journal of Marketing Communications*, 24(2):125–145, 2018.

[239] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *CoRR*, abs/1802.05365, 2018.

[240] Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, Véronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia Loukachevitch, Evgeniy Kotelnikov, Nuria Bel, Salud María Jiménez-Zafra, and Gülşen Eryiğit. SemEval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 19–30, San Diego, California, June 2016. Association for Computational Linguistics.

[241] Jim Powell. Amazon gets rid of "negative" votes on reviews. `https://www.financeafter50.com/amazon-gets-rid-of-negative-votes/`, October 2017.

[242] Aika Qazi, Karim Bux Shah Syed, Ram Gopal Raj, Erik Cambria, Muhammad Tahir, and Daniyal Alghazzawi. A concept-level approach to the analysis of online review helpfulness. *Computers in Human Behavior*, 58:75–81, 2016.

[243] Shanshan Qi, Cora Un In Wong, Ning Chen, Jia Rong, and Jiahua Du. Profiling Macau cultural tourists by using user-generated content from online social media. *Information Technology & Tourism*, 20(1):217–236, December 2018.

[244] Lingyun Qiu and Weiquan Wang. The effects of message order and information chunking on ewom persuasion. In *PACIS*, page 151, 2011.

[245] Xianshan Qu, Xiaopeng Li, and John R. Rose. Review helpfulness assessment based on convolutional neural network. *CoRR*, abs/1808.09016, 2018.

[246] Simon Quaschning, Mario Pandelaere, and Iris Vermeir. When consistency matters: The effect of valence consistency on review helpfulness. *Journal of Computer-Mediated Communication*, 20(2):136–152, 2015.

[247] Matthew Rabin and Joel L Schrag. First impressions matter: A model of confirmatory bias. *The quarterly journal of economics*, 114(1):37–82, 1999.

[248] Dragomir R. Radev, Hongyan Jing, Małgorzata Styś, and Daniel Tam. Centroid-based summarization of multiple documents. *Information Processing & Management*, 40(6):919–938, 2004.

[249] Colin Raffel and Daniel PW Ellis. Feed-forward networks with attention can solve some long-term memory problems. *arXiv preprint arXiv:1512.08756*, 2015.

[250] Randosity. Amazon's "not helpful" button missing? `https://randocity.com/2019/04/13/amazons-not-helpful-button-missing/`, April 2019.

[251] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. `http://is.muni.cz/publication/884893/en`.

[252] Demand Gen Report. The content preferences survey report. `https://www.demandgenreport.com/resources/research/the-2017-content-preferences-survey-report`, 2017.

[253] Filipe N. Ribeiro, Matheus Araújo, Pollyanna Gonçalves, Marcos

André Gonçalves, and Fabrício Benevenuto. Sentibench - a benchmark comparison of state-of-the-practice sentiment analysis methods. *EPJ Data Science*, 5(1):23, Jul 2016.

[254] Roman Rietsche, Daniel Frei, Emanuel Stöckli, and Matthias Söllner. Not all reviews are equal–a literature review on online review helpfulness. In *European Conference on Information Systems (ECIS)*, Stockholm, Sweden, June 2019. European Conference on Information Systems (ECIS).

[255] Ellen Riloff, Janyce Wiebe, and Theresa Wilson. Learning subjective nouns using extraction pattern bootstrapping. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 25–32. Association for Computational Linguistics, 2003.

[256] Everett M Rogers. *Diffusion of innovations*. Simon and Schuster, 2010.

[257] Jia Rong, Sandra Michalska, Sudha Subramani, Jiahua Du, and Hua Wang. Deep learning for pollen allergy surveillance from twitter in Australia. *BMC medical informatics and decision making*, 19(1):208, November 2019.

[258] Gobinda Roy, Biplab Datta, and Srabanti Mukherjee. Role of electronic word-of-mouth content and valence in influencing online purchase behavior. *Journal of Marketing Communications*, 0(0):1–24, 2018.

[259] Sebastian Ruder, Ivan Vulić, and Anders Søgaard. A survey of cross-lingual word embedding models. *Journal of Artificial Intelligence Research*, 65:569–631, 2019.

[260] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019.

[261] Mohammad Salehan and Dan J. Kim. Predicting the performance of online consumer reviews: A sentiment mining approach to big data analytics. *Decision Support Systems*, 81:30–40, 2016.

[262] Salesforce. The shopping index. `https://www.salesforce.com/solutions/industries/retail/shopping-index/`, 2019.

[263] Matthew J. Salganik, Peter Sheridan Dodds, and Duncan J. Watts. Experimental study of inequality and unpredictability in an artificial cultural market. *Science*, 311(5762):854–856, 2006.

[264] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manage.*, 24(5):513–523, August 1988.

[265] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, page 285–295, New York, NY, USA, 2001. Association for Computing Machinery.

[266] Sunil Saumya, Jyoti Prakash Singh, Abdullah Mohammed Baabdullah, Nripendra P. Rana, and Yogesh K. Dwivedi. Ranking online consumer reviews. *Electronic Commerce Research and Applications*, 29:78–89, 2018.

[267] Sunil Saumya, Jyoti Prakash Singh, and Yogesh K. Dwivedi. Predicting the helpfulness score of online reviews using convolutional neural network. *Soft Computing*, Feb 2019.

[268] Klaus R. Scherer. What are emotions? and how can they be measured? *Social Science Information*, 44(4):695–729, 2005.

[269] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015.

[270] Markus Schuckert, Xianwei Liu, and Rob Law. Insights into suspicious online ratings: Direct evidence from tripadvisor. *Asia Pacific Journal of Tourism Research*, 21(3):259–272, 2016.

[271] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681, 1997.

[272] United Parcel Service. Pulse of the online shopper: A customer experience study. `https://solutions.ups.com/rs/935-KKE-240/images/UPS-Pulse-of-the-Online-Shopper-Report.pdf`, 2019.

[273] Darren Shaw. Announcing the 2018 local search ranking factors survey. `https://moz.com/blog/2018-local-search-ranking-factors-survey`, November 2018.

[274] Dinghan Shen, Guoyin Wang, Wenlin Wang, Martin Renqiang Min, Qinliang Su, Yizhe Zhang, Chunyuan Li, Ricardo Henao, and Lawrence Carin. Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms. In *Proceedings of the 56th Annual Meeting of the Association*

for Computational Linguistics (Volume 1: Long Papers), pages 440–450, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[275] Seunghun Shin, Namho Chung, Zheng Xiang, and Chulmo Koo. Assessing the impact of textual content concreteness on helpfulness in online travel reviews. *Journal of Travel Research*, 58(4):579–593, 2019.

[276] Michael Siering, Jan Muntermann, and Balaji Rajagopalan. Explaining and predicting online review helpfulness: The role of content and reviewer-related signals. *Decision Support Systems*, 108:1–12, 2018.

[277] Stefan Siersdorfer, Sergiu Chelaru, Wolfgang Nejdl, and Jose San Pedro. How useful are your comments? analyzing and predicting youtube comments and comment ratings. In *Proceedings of the 19th international conference on World wide web*, pages 891–900, 2010.

[278] Riyaz T. Sikora and Kriti Chauhan. Estimating sequential bias in online reviews: A kalman filtering approach. *Know.-Based Syst.*, 27:314–321, March 2012.

[279] Jyoti Prakash Singh, Seda Irani, Nripendra P. Rana, Yogesh K. Dwivedi, Sunil Saumya, and Pradeep Kumar Roy. Predicting the "helpfulness" of online consumer reviews. *Journal of Business Research*, 70:346–355, 2017.

[280] Ruben Sipos, Arpita Ghosh, and Thorsten Joachims. Was this review helpful to you?: It depends! context and voting patterns in online content. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, pages 337–348, New York, NY, USA, 2014. ACM.

[281] Edgar A. Smith and J. Peter Kincaid. Derivation and validation of the automated readability index for use with technical materials. *Human Factors*, 12(5):457–564, 1970.

[282] Antoni Sobkowicz and Wojciech Stokowiec. Steam review dataset - new, large scale sentiment dataset. In J. Fernando Sánchez-Rada and Björn Schuller, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016) Workshop Emotion and Sentiment Analysis*, pages 55–58, Portorož, Slovenia, May 2016. European Language Resources Association (ELRA).

[283] Shrihari Sridhar and Raji Srinivasan. Social influence effects in online product ratings. *Journal of Marketing*, 76(5):70–88, 2012.

[284] Vartika Srivastava and Arti D. Kalro. Enhancing the helpfulness of online consumer reviews: The role of latent (content) factors. *Journal of Interactive Marketing*, 48:33–50, 2019.

[285] Statista. Number of digital buyers worldwide from 2014 to 2021. https://www.statista.com/statistics/251666/number-of-digital-buyers-worldwide/, July 2017.

[286] Shane Storks, Qiaozi Gao, and Joyce Y. Chai. Commonsense reasoning for natural language understanding: A survey of benchmarks, resources, and approaches. *CoRR*, abs/1904.01172, 2019.

[287] Carlo Strapparava and Alessandro Valitutti. WordNet affect: an affective extension of WordNet. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal, May 2004. European Language Resources Association (ELRA).

[288] Qi Su, Chu-Ren Huang, and Kai-yun Chen. Evidentiality for text trustworthiness detection. In *Proceedings of the 2010 Workshop on NLP and Linguistics: Finding the Common Ground*, pages 10–17, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

[289] Sudha Subramani, Sandra Michalska, Hua Wang, Jiahua Du, Yanchun Zhang, and Haroon Shakeel. Deep learning for multi-class identification from domestic violence online posts. *IEEE Access*, 7:46210–46224, April 2019.

[290] Xinyu Sun, Maoxin Han, and Juan Feng. Helpfulness of online reviews: Examining review informativeness and classification thresholds by search products and experience products. *Decision Support Systems*, 124:113099, 2019.

[291] Jiliang Tang, Huiji Gao, Xia Hu, and Huan Liu. Context-aware review helpfulness rating prediction. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, pages 1–8, New York, NY, USA, 2013. ACM.

[292] Jiliang Tang, Huiji Gao, Huan Liu, and Atish Das Sarma. Etrust: Understanding trust evolution in an online world. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, page 253–261, New York, NY, USA, 2012. Association for Computing Machinery.

[293] Mike Thelwall, Kevan Buckley, and Georgios Paltoglou. Sentiment strength detection for the social web. *J. Am. Soc. Inf. Sci. Technol.*, 63(1):163–173, January 2012.

[294] Alex S.L. Tsang and Gerard Prendergast. Is a "star" worth a thousand words?: The interplay between product-review texts and rating valences. *European Journal of Marketing*, 43(11/12):1269–1280, 2009.

[295] Oren Tsur and Ari Rappoport. Revrank: A fully unsupervised algorithm for selecting the most helpful book reviews. In *Third International AAAI Conference on Weblogs and Social Media*, 2009.

[296] TurnTo and Ipsos. Hearing the voice of the consumer. `http://www2.turntonetworks.com/2017consumerstudy`, March 2017.

[297] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, Berlin, Heidelberg, 1995.

[298] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[299] Chau Vo, Dung Duong, Duy Nguyen, and Tru Cao. From helpfulness prediction to helpful review retrieval for online product reviews. In *Proceedings of the Ninth International Symposium on Information and Communication Technology*, pages 38–45. ACM, 2018.

[300] Henning Wachsmuth, Martin Trenkmann, Benno Stein, Gregor Engels, and Tsvetomira Palakarska. A review corpus for argumentation analysis. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, pages 115–127, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[301] Joseph B. Walther, Yuhua (Jake) Liang, Tina Ganster, Donghee Yvette Wohn, and Josh Emington. Online Reviews, Helpfulness Ratings, and Consumer Attitudes: An Extension of Congruity Theory to Multiple Sources in Web 2.0. *Journal of Computer-Mediated Communication*, 18(1):97–112, 10 2012.

[302] Yun Wan. The matthew effect in online review helpfulness. In Jonna Järveläinen, Hongxiu Li, Anne-Marie Tuikka, and Tiina Kuusela, editors, *Co-created Effective, Agile, and Trusted eServices*, pages 38–49, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[303] Yun Wan. The matthew effect in social commerce. *Electronic Markets*, 25(4):313–324, 2015.

[304] Yun Wan and Makoto Nakayama. Are amazon.com online review helpfulness

ratings biased or not? In Michael J. Shaw, Dongsong Zhang, and Wei T. Yue, editors, *E-Life: Web-Enabled Convergence of Commerce, Work, and Social Life*, pages 46–54, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[305] Erin Yirun Wang, Lawrence Hoc Nang Fong, and Rob Law. Review helpfulness: The influences of price cues and hotel class. In Julia Neidhardt and Wolfgang Wörndl, editors, *Information and Communication Technologies in Tourism 2020*, pages 280–291, Cham, 2020. Springer International Publishing.

[306] Hongning Wang, Yue Lu, and ChengXiang Zhai. Latent aspect rating analysis without aspect keyword supervision. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, page 618–626, New York, NY, USA, 2011. Association for Computing Machinery.

[307] Richard Y Wang and Diane M Strong. Beyond accuracy: What data quality means to data consumers. *Journal of management information systems*, 12(4):5–33, 1996.

[308] Ting Wang, Dashun Wang, and Fei Wang. Quantifying herding effects in crowd wisdom. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 1087–1096, New York, NY, USA, 2014. ACM.

[309] Yani Wang, Jun Wang, and Tang Yao. What makes a helpful online review? a meta-analysis of review characteristics. *Electronic Commerce Research*, Jun 2018.

[310] Jason Wei and Kai Zou. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. 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)*, pages 6383–6389, Hong Kong, China, November 2019. Association for Computational Linguistics.

[311] John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. Towards universal paraphrastic sentence embeddings. *arXiv preprint arXiv:1511.08198*, 2015.

[312] John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. Towards universal paraphrastic sentence embeddings. *arXiv preprint arXiv:1511.08198*, 2015.

[313] Lotte M. Willemsen, Peter C. Neijens, Fred Bronner, and Jan A. de Ridder. "highly recommended!" the content characteristics and perceived usefulness

of online consumer reviews. *Journal of Computer-Mediated Communication*, 17(1):19–38, 2011.

[314] Jianan Wu. Review popularity and review helpfulness: A model for user review effectiveness. *Decision Support Systems*, 97:92–103, 2017.

[315] Philip Fei Wu, Hans Van Der Heijden, and Nikolaos Korfiatis. The influences of negativity and review quality on the helpfulness of online reviews. In *International conference on information systems*, pages 1–10, August 2011.

[316] Zheng Xiang, Qianzhou Du, Yufeng Ma, and Weiguo Fan. A comparative analysis of major online review platforms: Implications for social media analytics in hospitality and tourism. *Tourism Management*, 58:51–65, 2017.

[317] Hong Xie, Yongkun Li, and John CS Lui. Understanding persuasion cascades in online product rating systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5490–5497, 2019.

[318] Wenting Xiong and Diane Litman. Automatically predicting peer-review helpfulness. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, HLT '11, pages 502–507, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[319] Wenting Xiong and Diane Litman. Empirical analysis of exploiting review helpfulness for extractive summarization of online reviews. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1985–1995, Dublin, Ireland, August 2014. Dublin City University and Association for Computational Linguistics.

[320] Sung-Byung Yang, Seung-Hun Shin, Youhee Joun, and Chulmo Koo. Exploring the comparative importance of online hotel reviews' heuristic attributes in review helpfulness: a conjoint analysis approach. *Journal of Travel & Tourism Marketing*, 34(7):963–985, 2017.

[321] Y. Yang, C. Chen, and F. S. Bao. Aspect-based helpfulness prediction for online product reviews. In *2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 836–843, Nov 2016.

[322] Yinfei Yang, Yaowei Yan, Minghui Qiu, and Forrest Bao. Semantic analysis and helpfulness prediction of text for online product reviews. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the*

7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pages 38–44, Beijing, China, July 2015. Association for Computational Linguistics.

[323] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, San Diego, California, June 2016. Association for Computational Linguistics.

[324] Yelp. Yelp dataset challenge. `https://www.yelp.com/dataset/challenge`, January 2019.

[325] Dezhi Yin, Samuel D. Bond, and Han Zhang. Anxious or angry? effects of discrete emotions on the perceived helpfulness of online reviews. *MIS Quarterly*, 38(2):539–560, June 2014.

[326] Dezhi Yin, Sabyasachi Mitra, and Han Zhang. Research note—when do consumers value positive vs. negative reviews? an empirical investigation of confirmation bias in online word of mouth. *Information Systems Research*, 27(1):131–144, 2016.

[327] Wenpeng Yin, Katharina Kann, Mo Yu, and Hinrich Schütze. Comparative study of cnn and rnn for natural language processing. *arXiv preprint arXiv:1702.01923*, 2017.

[328] Kyung-Hyan Yoo and Ulrike Gretzel. Comparison of deceptive and truthful travel reviews. In Wolfram Höpken, Ulrike Gretzel, and Rob Law, editors, *Information and Communication Technologies in Tourism 2009*, pages 37–47, Vienna, 2009. Springer Vienna.

[329] Lin Yue, Weitong Chen, Xue Li, Wanli Zuo, and Minghao Yin. A survey of sentiment analysis in social media. *Knowledge and Information Systems*, 60(2):617–663, Aug 2019.

[330] Lin Yue, Weitong Chen, Xue Li, Wanli Zuo, and Minghao Yin. A survey of sentiment analysis in social media. *Knowledge and Information Systems*, 60(2):617–663, Aug 2019.

[331] Yi-Ching Zeng, Tsun Ku, Shih-Hung Wu, Liang-Pu Chen, and Gwo-Dong Chen. Modeling the helpful opinion mining of online consumer reviews as a classification problem. *International Journal of Computational Linguistics & Chinese Language Processing, Volume 19, Number 2, June 2014*, 19(2), 2014.

[332] Liyi Zhang, Daomeng Guo, Xuan Wen, Yiran Li, et al. Effect of other visible reviews' votes and personality on review helpfulness evaluation: an event-related potentials study. *Electronic Commerce Research*, pages 1–25, May 2020.

[333] Richong Zhang and Thomas Tran. An information gain-based approach for recommending useful product reviews. *Knowledge and Information Systems*, 26(3):419–434, 2011.

[334] Richong Zhang, Thomas Tran, and Yongyi Mao. Opinion helpfulness prediction in the presence of "words of few mouths". *World Wide Web*, 15(2):117–138, Mar 2012.

[335] Xiaoying Zhang, Junzhou Zhao, and John C.S. Lui. Modeling the assimilation-contrast effects in online product rating systems: Debiasing and recommendations. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, RecSys '17, pages 98–106, New York, NY, USA, 2017. ACM.

[336] Ye Zhang and Byron Wallace. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 253–263, Taipei, Taiwan, November 2017. Asian Federation of Natural Language Processing.

[337] Z. Zhang, J. Qi, and G. Zhu. Mining customer requirement from helpful online reviews. In *2014 Enterprise Systems Conference*, pages 249–254, Aug 2014.

[338] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. ERNIE: enhanced language representation with informative entities. *CoRR*, abs/1905.07129, 2019.

[339] Zhu Zhang and Balaji Varadarajan. Utility scoring of product reviews. In *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, CIKM '06, pages 51–57, New York, NY, USA, 2006. ACM.

[340] Zunqiang Zhang, Yue Ma, Guoqing Chen, and Qiang Wei. Extending associative classifier to detect helpful online reviews with uncertain classes. In *Proceedings of the 2015 Conference of the International Fuzzy Systems Association and the European Society for Fuzzy Logic and Technology*, pages 1134–1139, June 2015.

[341] Pengfei Zhao, Ji Wu, Zhongsheng Hua, and Shijian Fang. Finding eWOM customers from customer reviews. *Industrial Management & Data Systems*, 119(1):129–147, 2019.

[342] Lei Zheng, Vahid Noroozi, and Philip S. Yu. 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, pages 425–434, New York, NY, USA, 2017. ACM.

[343] Xiaolin Zheng, Shuai Zhu, and Zhangxi Lin. Capturing the essence of word-of-mouth for social commerce: Assessing the quality of online e-commerce reviews by a semi-supervised approach. *Decision Support Systems*, 56:211–222, 2013.

[344] Chunting Zhou, Chonglin Sun, Zhiyuan Liu, and Francis Lau. A C-LSTM neural network for text classification. *arXiv preprint arXiv:1511.08630*, 2015.

[345] Shasha Zhou and Bin Guo. The interactive effect of review rating and text sentiment on review helpfulness. In Heiner Stuckenschmidt and Dietmar Jannach, editors, *E-Commerce and Web Technologies*, pages 100–111, Cham, 2015. Springer International Publishing.

[346] Shasha Zhou and Bin Guo. The order effect on online review helpfulness. *Decision Support System*, 93:77–87, January 2017.

[347] Yusheng Zhou and Shuiqing Yang. Roles of review numerical and textual characteristics on review helpfulness across three different types of reviews. *IEEE Access*, 7:27769–27780, 2019.

[348] Yusheng Zhou, Shuiqing Yang, Yixiao Li, Yuangao chen, Jianrong Yao, and Atika Qazi. Does the review deserve more helpfulness when its title resembles the content? locating helpful reviews by text mining. *Information Processing & Management*, 57(2):102179, 2020.

[349] Ling Zhu, Guopeng Yin, and Wei He. Is this opinion leader's review useful? peripheral cues for online review helpfulness. *Journal of Electronic Commerce Research*, 15(4):267, 2014.