Spring 5-2021

HETEROGENEOUS GRAPH-BASED USER-SPECIFIC REVIEW HELPFULNESS PREDICTION

Dongkai Chen
Dongkai.Chen.GR@Dartmouth.edu

HETEROGENEOUS GRAPH-BASED USER-SPECIFIC REVIEW HELPFULNESS PREDICTION

A Thesis

Submitted to the Faculty

in partial fulfillment of the requirements for the

degree of

Master of Science

in

Computer Science

by

Dongkai Chen

DARTMOUTH COLLEGE

Hanover, New Hampshire

May 2021

Examining Committee:

Venkatramanan Siva Subrahmanian, Chair

Soroush Vosoughi

Mauro Conti

F. Jon Kull, Ph.D
Dean of the Guarini School of Graduate and Advanced Studies

Abstract

With the popularity of e-commerce and review websites, it is becoming increasingly important to identify the helpfulness of reviews. However, existing works on predicting reviews' helpfulness have three major issues: (i) the correlation between helpfulness and features from review text is not clear yet, although many standard features are proposed, (ii) the relations between users, reviews and products have not been considered, (iii) the effectiveness of the existing approaches have not been systematically compared. To address these challenges, we first analyze the correlation between standard features and review helpfulness that are widely used in other work. Based on this analysis, we propose an end-to-end neural network architecture, the Global-Local Heterogeneous Graph Neural Networks (GL-HGNN). It consists of the graph construction and learning nodes representations both globally and locally. The graph is composed of three types of nodes including users, reviews and products, as well as four link types to build connections among these nodes. To better learn the feature representations, we employ a global graph neural network (GNN) branch and a local GNN branch on the whole graph and associated subgraphs to capture graph structure and information propagation. Finally, we provide an empirical comparison with traditional machine learning models training on hand-crafted features as well as four state-of-the-art deep learning models on eight Amazon product categories.

Preface

The thesis is the original work of the author Dongkai Chen. The work presented was conducted in the Dartmouth Security and AI lab.

Acknowledgements

Throughout the journey of my master and the process of starting research, I am grateful to receive lots of invaluable guidance and support from my advisor, colleagues and friends.

I'd like to express gratitude to my advisor, Prof. V.S. Subrahmanian, for his guidance and big support over the two years life at Dartmouth. I still remember the beginning of this trip. I felt so excited when Prof. V.S was willing to let me join DSAIL. I then worked with Yanhai Xiong to start my first project. I learned a lot during that time. After that, I participated other work and worked with other excellent and smart people. Prof. V.S is always very supportive and helpful. Thank you for bringing me into the door of the state-of-the-art machine learning research and providing me with practical suggestions during research. Your rigorous research attitude deeply affects me and motivates me to do a great research.

I would also like to thank my collaborators and labmates: Chongyang Bai, Haipeng Chen, Yanhai Xiong, Luca Pajola. It is your engagement, encouragement and valuable research ideas that offer me the motivation.

I really enjoyed my life in Hanover, a beautiful small town. I can swim in a lake in the hot summer, enjoy the beautiful forest in the autumn and ski in the winter.

Thanks Dartmouth!

Last but not least, I want to say thank you to my family and my girl friend for their support and encouragement. I can always keep going because of your support and love.

Thank all of the people in my life again!

Contents

Chapter 1

Introduction

Information technology has brought many benefits to our society, people can easily buy anything they want through online shopping platform without going outside. More and more people leave their comments on their purchased products on the website, which helps the other customers to determine whether this product is worth. Thus, it also increases the sale of the product[1]. However, due to the explosive growth of reviews on websites, it is hard to identify useful reviews from tens of thousands of reviews in a short period of time. The votes of these reviews follows a significant long tail distribution due to the current voting process in online e-commerce websites [2]. Since the reviews that first receive more votes will continuously capture more users attention compared to those reviews having fewer votes. Similar phenomenon is also found in low-traffic products, only a few customers will purchase and recent products [3].

Amazon is one of the largest e-commerce website and there are millions of reviews on their platform. Users in *Amazon* will upvote a review if they think it is a helpful review. Although there is a bias in voting process, past work considers these votes as human labeled labels. To be more specific, some work consider a review is helpful if the ratio of helpful votes to total votes is larger than or equal to 0.75 [2, 3, 4]. Thus,

this problem can be modeled as a supervised classification task.

To the best of our knowledge, the previous studies mainly focus on two parts.

(a) To extract different kinds of hand-crafted features including lexical, structural, syntactic features and features from meta data based on data analysis and statistics [5, 6, 7, 8, 9, 10, 11].

(b) To use various neural networks, e.g, convolutional neural network, LSTMs and attention mechanism to learn the embedding representation of review text followed by a linear layer for the prediction [3, 12, 13].

However, there are still three problems remaining to be solved. 1) Although there are various standard features proposed by researchers from different domains, most of them lack statistical analysis. For example, why these features are helpful for models prediction? Are these features suitable for different datasets? 2) Most work including hand-crafted features construction and neural network approaches extract features from text and meta data. Training a good embedding of text is also a feature extraction method, while they ignore the relation between reviews and users, reviews and products, users with other users who have similar shopping experience. 3) There is no empirical comparisons between these approaches. One reason is that different approaches have been trained on different datasets and not all of them are public for reproduction. Another reason is that new features proposed by researcher are not well evaluated and new methods are not carefully compared with baselines.

In this paper, we investigate the problem of review helpfulness prediction. Based on graph research, together with past work on fraudulent user prediction [14] and vulnerabilities prediction [15], this thesis has several contributions. First, we conduct a deep analysis on the review helpfulness prediction problem and standard features proposed by other studies. Second, we propose Global-Local Heterogeneous Graph Neural Networks (GL-HGNN) framework. We model relations of users, reviews and

products by constructing a heterogeneous graph, named user-review-product graph (URP graph) and then employ GNNs on the entire user-review-product graph and associated subgraphs including user-review graph, review-product graph and product-product graph extracted from the graph to learn nodes representations globally and locally. Lastly, we evaluate our approach on 8 Amazon product categories and conduct a fair comparison between our method and popular baselines including traditional machine learning models training with hand-crafted features and deep neural networks.

Chapter 2

Related Work

Ocampo et al. conducted a comprehensive survey on helpfulness prediction which categorize mainstream methods into hand-crafted features and embedding features [2]. Based on recent neural network methods [3], we categorize current approaches for predicting review helpfulness into hand-crafted feature based approaches and neural network based approaches. We summarize the state-of-the-art approaches in Table 2.1, which shows the features involved for prediction.

Section 2.1

Prediction using Hand-crafted Features

(a) **Structural Features** [5, 6]: Structural features usually include statistical features, e.g., number of tokens, number of sentences in a review, average sentences length and HTML tag which exists in data crawling from website.

(b) **Lexical Features** [5, 7]: Lexical features refers to features like N-grams and spelling errors. However, because of the massive text data, even 2-grams requires huge memory for storage. In practice, $tf - idf$ is used as replaced methods for filtering those low frequency n-grams.

		[5]	[16]	[6]	[7]	[8]	[10]	[17]	[18]	[19]	[3]	[12]	[13]
Structural Features	# Tokens	x		x	x		x	x					
	# Sentences	x		x	x		x	x					
	Avg. Sen. Length	x		x	x		x	x					
Lexical Features	HTML Tag	x		x									
	Ngrams Spelling errors	x						x					
Syntactic Features	Summary Statistics	x						x					
Semantic Features	Product-Features	x											
	Sentiment Score	x						x					
	Emotions							x					
	Experience								x				
	Readability							x					
Meta Data	Overall Scores	x		x	x	x	x	x		x			
	Users Context		x			x							
	Product Reviews				x		x			x			
	Temporal				x	x	x			x			
	Product Type				x								
	Product Info									x	x		
DNN											x	x	x

Table 2.1: Literature review. This table summarize different features used in different work.

(c) **Syntactic Features** [5, 7]: Syntactic features extract Part of Speech tag for each token in a review. Available features can be derived from the number/percentage of tokens, which are nouns, adjectives, adverbs, etc.

(d) **Semantic Features** [5, 7, 8]: Readability score measures the readability of the review. Emotions of a review/sentence in a review can be used for evaluating users' attitude toward the products. [8] also use subjectivity to evaluate the objectiveness of the review, i.e., if the review describe both the advantages of the product and its disadvantages. Other features such as sentiment score and experience of a user is also used in past work.

(e) **Meta Data**: Many studies introduce features from meta data which contain product related information, the ratings of the product and temporal information. In study [6, 7], they use overall scores of products as one of the features which shows a positive correlation between star rating and helpfulness by [20]. Chen et al. [21] introduces past reviews of users and past helpful votes as context

features while [16] considers the user-reviewer connection strength in a social network. Moreover, temporal information is also taken into account in [22, 21, 23], product-related information like product type and product title are also used in [9, 3].

Section 2.2

Prediction using Neural Networks

Deep learning has achieved great success in computer vision [24] and natural language processing [25]. With the help of deep learning technique, we do not have to spend too much time to manually design domain-specific features and heuristic algorithm to extract features from text. Several works design various architectures for better predicting review helpfulness to avoid tedious feature engineering work [3, 12, 13]. The work in [12, 13] use convolutional neural network model to extract text features, i.e., to learn the embedding representation of review text. Furthermore, different embedding techniques may contribute to different results. To control the word embedding fed into downstream models, Chen et al.[12] uses word-level embedding-gates while Devlin et al. [25] uses sub-word embedding technique on text classification tasks. Due to the power of attention mechanism in other domains, e.g., machine translation [26] and powerful pre-trained model [25, 27], Fan et al. designed a neural network to jointly learn the embedding of the review text and the product title, and use an attention mechanism to learn how much information that the review text can benefit from product title [3].

Chapter 3

Dataset Statistics

Our dataset consists of eight product categories of reviews including product reviews and metadata from Amazon (May 1996 - July 2014). The datasets are collected from the Amazon platform with different products domains. Each valid data sample consists of the following components: a user profile of who bought this product and wrote the review, a review text, a rating of this product and the product information in meta data. Table 3.1 shows overall statistics of the 8 datasets which includes the number of products, reviews, users in each category. In total, there are 941214 reviews in our all datasets.

There is no ground truth of the helpfulness for the reviews, we only have the

Dataset	# of Products	# of Reviews	# of Users
Clothing Shoes & Jewelry	15044	30995	17693
Grocery & Gourmet Food	7115	24736	8783
Health & Personal Care	15426	69351	25469
Home & Kitchen	23937	109556	40081
Movies & TV	48709	633719	84833
Pet Supplies	6189	17839	8873
Tools & Home Improvement	8496	29274	10337
Toys & Games	9119	25744	9434

Table 3.1: Statistics of the 8 Amazon categories. We list the number of products, reviews and users of the 8 different categories.

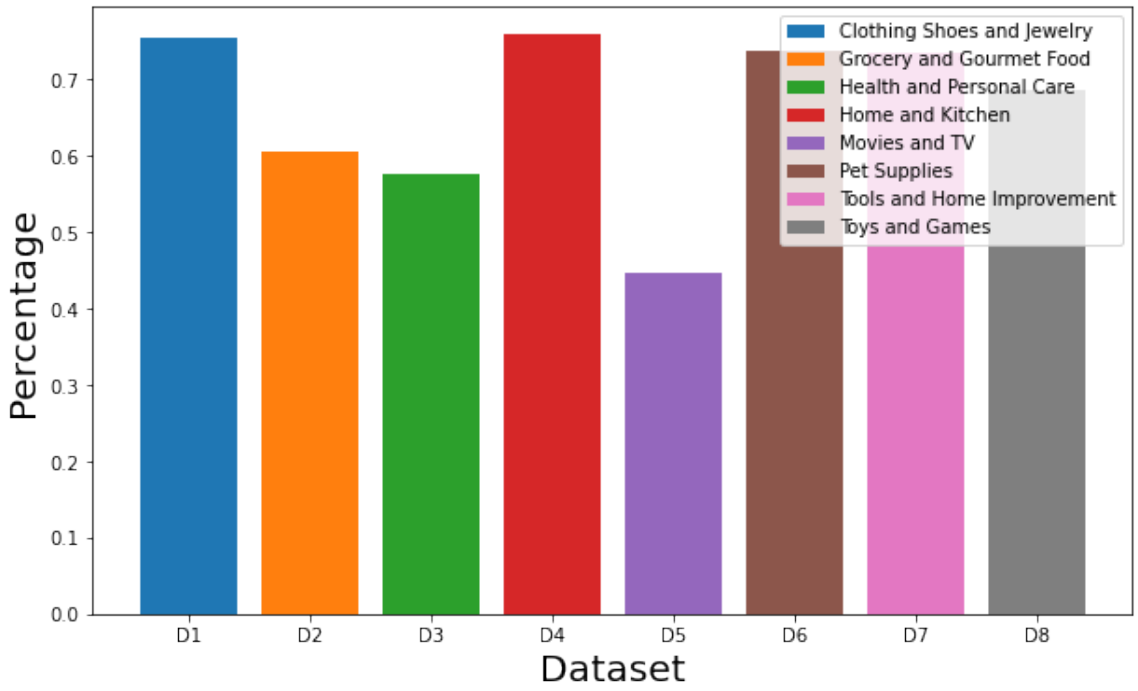

Figure 3.1: Percentage of positive samples in each category

number of upvotes and total votes. We follow [2] to process the labels of each review. To be more specific, we mark the reviews as positive samples (helpful reviews) if the review receives at least 75% of helpful votes with respect to the total votes. The rest of them are regarded as negative samples. Figure 3.1 shows the ratio of positive samples in each dataset. We can see that the percentage of positive samples in *Clothing Shoes and Jewelry*, *Home and Kitchen*, *Pet Supplies* and *Tools and Home Improvement* are very high (close to 75%) while *Movies and TV* only have 44.6% positive samples. This indicates it is an unbalanced dataset. To investigate the label imbalance problem, we first sort all reviews in each dataset according to the number of votes that each review received, then split the whole dataset into 10 partitions. We show the change of percentage of positive samples from 1 to 10 partition in Figure 3.2. It shows a clear increasing trend indicating that there are more positive samples when the number of votes increases, compared with the positive samples percentage of the last partition

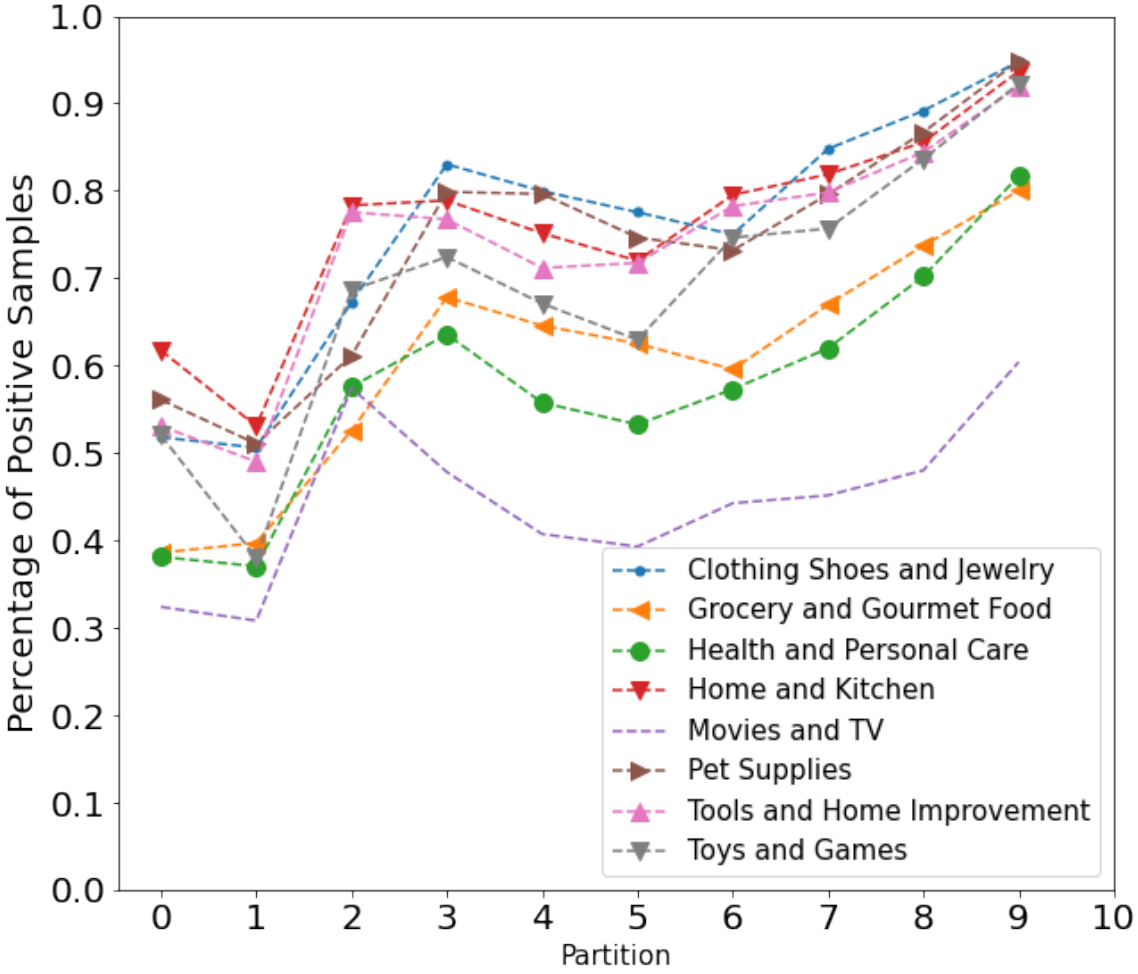

Figure 3.2: The change of percentage of positive samples in different partitions. The x-axis denotes the partition number from 1 to 10, the y-axis is the percentage of positive samples in each partition.

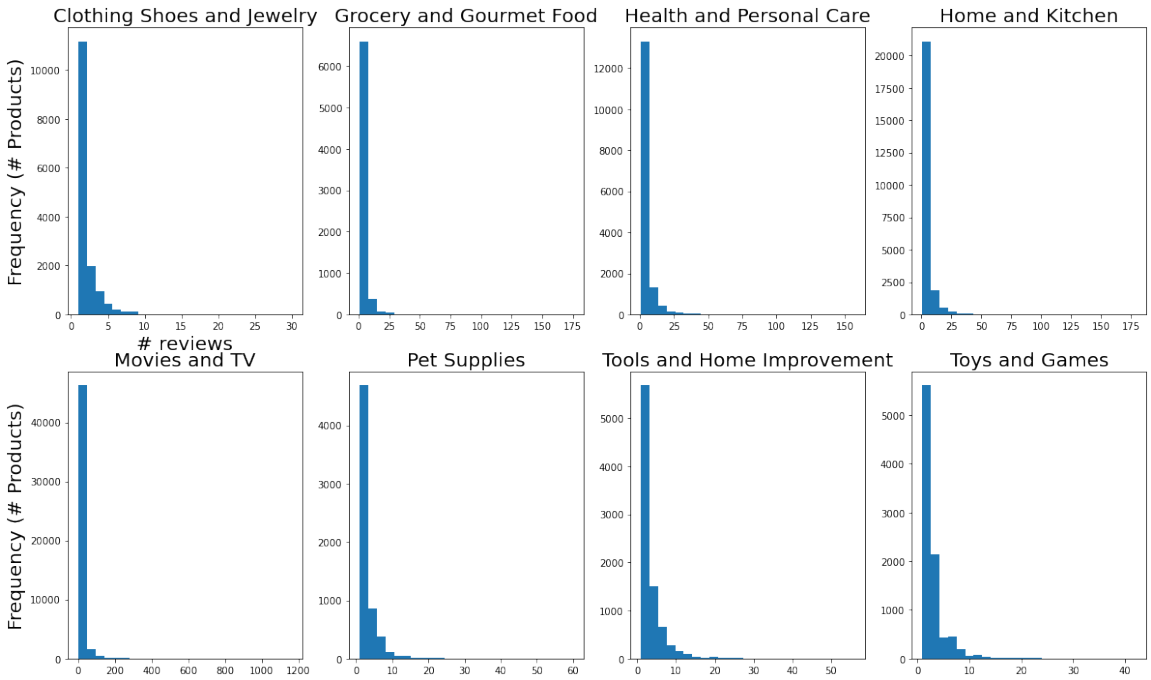

Figure 3.3: Distribution of products having different number of reviews. The X-axis denotes the number of reviews and the Y-axis denotes the number of products.

with percentage of the first partition.

We also study the difference of number of reviews in different products. Figure 3.3 shows the distribution of the number of reviews of different products. The x-axis denotes the number of reviews and the Y-axis denotes the number of products. All the 8 sub-figures shows the long-tail distribution indicating that most of products only have a few reviews, which leads to difficulty in predicting reviews helpfulness.

Chapter 4

Features Analysis

One contribution of this paper is to investigate which features in the machine learning models contribute more to predict review helpfulness. Table 4.1 shows the correlation between five types of features (structural, lexical, semantic features, meta data and context features) and review helpfulness.

Structural Features. Structural features denote text structure and formatting which shows the structural complexity. Here we consider three features, the number of tokens, the number of sentences and the average sentence length.

Lexical Features. Lexical features are extracted from word-level which include n-grams and number of spelling errors.

Semantic Features. We consider two types of semantic features, the readability score and the emotion entropy.

(a) Readability score. It measures the difficulty of reading a text. State-of-the-art work uses multiple readability measures, e.g., Gunning Fog Index (GFI) and Automated Readability Index (ARI), which are defined as follows:

$$GFI = 0.4(\frac{\#\text{ words}}{\#\text{ sentences}} + 100\frac{\#\text{ complex words}}{\#\text{ words}}) \tag{4.1}$$

$$ARI = 4.71\frac{\#\text{ characters}}{\#\text{ words}} + 0.5\frac{\#\text{ words}}{\#\text{ sentences}} - 21.43 \tag{4.2}$$

(b) Emotion entropy. We extract emotions of each sentence in a review using the text2emotion approach [1], and calculate the entropy of emotions following [20]. The definition of emotion entropy is as follows:

$$\text{Emotion Entropy} = \sum_{sent \in review} \text{Extractor(sent)} \log \text{Extractor(sent)} \tag{4.3}$$

where the extractor receiving a sentence from the review output an emotion, e.g., happiness, ranging from 0 (unhappy) to 1 (happy).

Meta Data. Extracting features from meta data is also a mainstream approach to predict review helpfulness. We explore the following features from Amazon review meta data.

(a) Number of Votes: A user can upvote/downvote a review if he thinks it is helpful or not. We use the number of votes as one of the features indicating how many people vote on this review.

(b) Overall Score: We use the overall score as our feature since a user may not write a review after buying a product but left a rating score to this product. The range of the score is 1 to 5.

Context Features. Contextual features reveal the relationship between users, reviews and products. For example, two users with similar shopping patterns will leave similar reviews that have similar helpfulness scores, and two products with similar reviews will lead to the same user response if they have similar reviews. Then, we explore the following features.

[1] https://shivamsharma26.github.io/text2emotion/

(a) User Average Ratings: A user may have multiple ratings scores for different products. We calculate the average score of each user past ratings as follows:

$$\text{Average Ratings} = \frac{1}{N}\sum_{i \in N} rating_i \tag{4.4}$$

where N is the total number of reviews that the user has, $rating_i$ indicates the rating of i th review of this user.

(b) Number of Past Reviews: We compute the total number of reviews of the product.

(c) Product Rating Discrepancy: For each review of the product, we compute the distance between the review's rating of this product and the average past rating scores of this product. The formulation is as follows:

$$\text{Discrepancy} = rating_i - \frac{1}{M}\sum_{j \in M} rating_j \tag{4.5}$$

where $rating_i$ indicates the rating of ith review of this user, M denotes the number of reviews before the ith review and we calculate the average ratings of these reviews.

We then employ the Pearson correlation coefficient and p-value for testing non-correlation. The correlation coefficient is calculated as follows:

$$r = \frac{\sum(x - m_x)(y - m_y)}{\sqrt{\sum(x - m_x)^2 \sum(y - m_y)^2}} \tag{4.6}$$

where m_x is the mean of the vector x and m_y is the mean of the vector y.

Table 4.1 shows the Pearson correlation coefficient which measures the relationship between the above features and review helpfulness. Although we believe some features

		Clothing Shoes & Jewelry	Grocery & Gourmet Food	Health & Personal Care	Home & Kitchen	Movies TV	Pet Supplies	Tools & Home Improvement	Toys Games
Structural Features	# Tokens	0.054	0.032	0.123	0.098	0.168	0.114	0.134	0.102
	# Sentences	0.055	0.032	0.121	0.095	0.129	0.091	0.128	0.093
	Avg. Length	-0.019	-0.003	0.038	0.024	0.055	0.02	0.028	0.018
Semantic Features	ARI	0.028	0.008	0.079	0.062	0.093	0.061	0.089	0.067
	GFI	0.028	0.008	0.078	0.061	0.092	0.061	0.088	0.065
	Emotion	0.078	0.039	0.116	0.103	0.182	0.127	0.121	0.105
Lexical Features	# Mistakes	0.029	0.01	0.091	0.078	0.126	0.087	0.113	0.084
Meta Data	Overall Scores	0.242	0.4	0.288	0.286	0.375	0.344	0.283	0.25
	# Voter	0.13	0.081	0.077	0.034	0.096	0.091	0.119	0.124
Context Features	User Average Ratings	0.038	0.109	0.076	0.075	0.25	0.067	0.105	0.131
	# Past Reviews	-0.06	-0.202	-0.09	-0.107	-0.182	-0.145	-0.082	-0.106
	Product Rating Discrepency	0.165	0.29	0.227	0.223	0.283	0.265	0.199	0.201

Table 4.1: Linear correlation between features and review helpfulness evaluated on 8 Amazon categories. Note that all correlation are statistically significant ($p < 0.05$).

should play important roles in predicting helpfulness, they show no positive correlation with helpfulness. Structural features have low correlation with helpfulness in some datasets, e.g., *Health & Personal Care* and *Movies & TV*. Readability scores do not reflect correlation with helpfulness while emotion entropy does in 5 out of 8 datasets. Number of votes in meta data shows strong correlation compared to other features. Finally, we observe that context features show high correlation with helpfulness, especially for the product rating discrepancy. Based on the findings from the table, we further explore context features. We not only consider relation between reviews and products, but also users and reviews, products and products. In this case, we use a graph to represent their connections. The nodes in the graph indicate the users, reviews and products, and edges represent their relationship. In the following chapter, we explain the graph construction and how to use the graph to predict review helpfulness.

Chapter 5

Methodology

In this chapter, we first describe the problem of review helpfulness prediction and introduce the fundamental concepts and notations in this article. After that, we present the proposed methodology to predict review helpfulness. Table 5.1 summarizes the notations used in this thesis.

Section 5.1

Preliminary

Predicting Review Helpfulness. Given a review of a product, the task is to predict the helpfulness of this review. We model this problem as a supervised problem, the raw input includes the following information: users that write the review ($U = u_1, u_2, ...$), reviews content ($R = r_1, r_2$) and the products ($P = p_1, p_2, ...$). The output is denoted as a label $Y \in [0, 1]$, which indicates if the current review is helpful or not. Assuming that we would like to predict the helpfulness of a single review, the problem can be formalized as the following objective function:

$$arg\ min_\theta L(F(\theta, r, p, u), Y) \tag{5.1}$$

Table 5.1: Notations used in the paper

Notations	Meanings
$U = \{u_1, u_2, ...\}$	the set of all users
$R = \{r_1, r_2, ...\}$	the set of all reviews
$P = \{p_1, p_2, ...\}$	the set of all products
$G = (U, P, R)$	the graph constructed from $\{U, R, P\}$
$G^{U,R} = (U, R)$	the graph constructed from $\{U, R\}$
$G^{R,P} = (R, P)$	the graph constructed from $\{R, P\}$
$G^{P,P} = (P, P)$	the graph constructed from $\{P, P\}$

where L is the loss function, e.g., cross-entropy loss function for classification problem, F is a model for prediction and θ denotes model parameters. The goal of this equation is to find the best parameters θ which minimizes the loss function, in other words, to better predict the review helpfulness.

Heterogeneous graph. A heterogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E},$ $O_V, R_E)$ consists of a vertex set $\mathcal{V}$, a link set $\mathcal{E}$. O_V and R_E represent the types of the object and the types of the link relation, respectively.

Section 5.2

The Heterogeneous Graph Construction

As discussed in Chapter 4, features extracted from users' past experience and products have positive correlation with review helpfulness. Therefore, we attempt to construct a graph G where we build bridges among users, reviews and products.

As shown in Figure 5.1, we construct a heterogeneous graph to model the relations among users, reviews and products. This graph consists of three types of objects (Users (U), Reviews (R), Products (P)) and five types of link relations (one between users and reviews, one between reviews and products, and the other three in products). From a review sample, we know the review content (r) and who wrote this review (u), which leads to a connection between a review node and a user node. A user can write

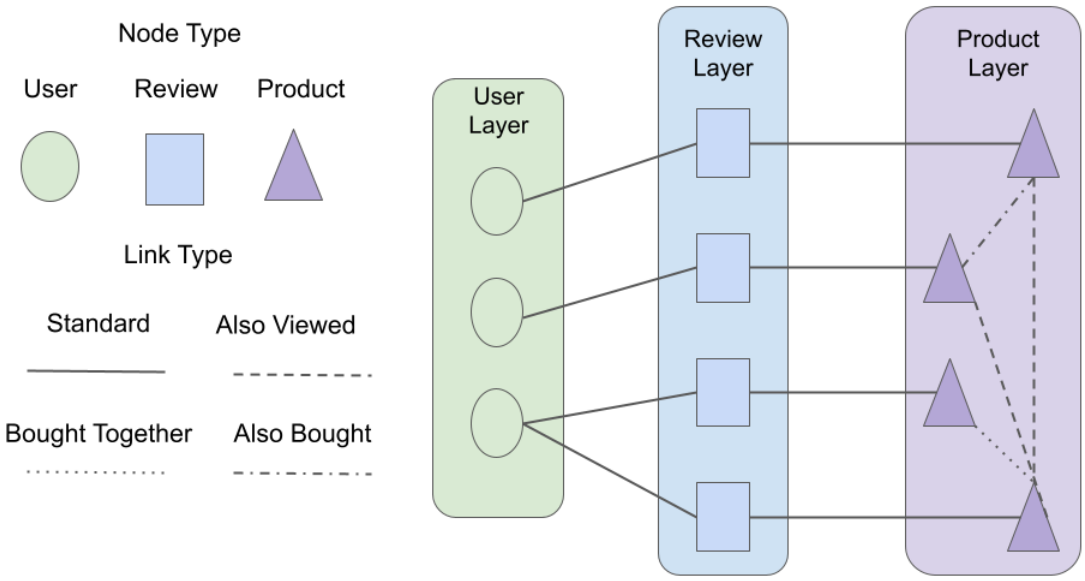

Figure 5.1: The User-Review-Product Graph. This heterogeneous graph includes three type of nodes which can be categorized as the user layer, review layer and product layer. There are total four link types in this graph, one connection type exists in user-review graph, one is in review-product graph and the other three exist in products and their neighbours (product-product graph).

multiple reviews but a review only belongs to one user. We also know the product from the review. Therefore, there is a link between the review node and the product node. By leveraging the information from meta data, we know three different link relations among products. (i) *Also Viewed*. This denotes that a user u also viewed a product p_2, e.g., iPhone 10, after viewing the product p_1, e.g., iPhone 8, which means the p_1 node will have a connection with p_2 representing that they may share similar characteristics which attract the user (ii) *Also Bought*. User u who bought an iPhone 10 also bought Airpods and IPads previously, so there is another link type between these products. (iii) *Bought Together*. The user u decided buy both the iPhone and the phone case. In this case, the third link type will be used to connect the iPhone and phone case.

We do not have a connection between user layers because there is no clear relationship between them. We do not know if two users know each other or if they are in the same age group, but we can infer their shopping preferences by referring to their past experiences, the reviews they write, and the products they buy. Thus, even if we do not explicitly link user nodes, information can be propagated through the review and product layers. There are no edges between review nodes, and although we can explicitly link two reviews when their content is similar, calculating the distance between two reviews takes a lot of time, e.g. we need to first extract all the review features and enumerate the distance of each review from the other reviews.

We divide the heterogeneous graph into three sub-graphs: the user-review graph (U-R graph), the review-product graph (R-P graph) and the product-product graph (P-P graph).

(a) **User-Review Graph**. Users' nodes are connected to the reviews they write. Each user may be connected to multiple reviews, but a review belongs to only one user.

(b) **Review-Product Graph**. Product nodes are linked to the reviews they are associated with. A product can have zero to multiple reviews, but a review belongs to only one specific product.

(c) **Product-Product Graph**. As we discussed above, products are intrinsically linked to other related products, and there are three link types in this graph, representing also seen, also bought, and bought together.

(d) **User-Review-Product Graph**. It combines the above three graphs together and creates a bridge between the three types of nodes. Although there are no internal connections in the user and review layers, mutual information can be propagated through the product layer. For example, a user node can access another user node if both users write a review for the same product, which means that at least one path in the graph is possible.

Section 5.3

The GL-HGNN Framework

After the graph construction, we employee the graph neural networks to learn features from the graph both globally and locally. More specifically, there are two GNNs branches, the global GNN branch is responsible for learning representations of all the three types nodes in the graph, and the other local GNN branch including multiple GNN blocks is employed on part of the graph for fine-grained feature representation, there are GNN blocks on the user-review graph, review-product graph and product-product graph separately.

The following chapter presents details of our approach. We explain how we update the parameters in the model and feature representations of nodes during training.

Figure 5.2 shows the architecture of the GL-HGNN. There are two branches of

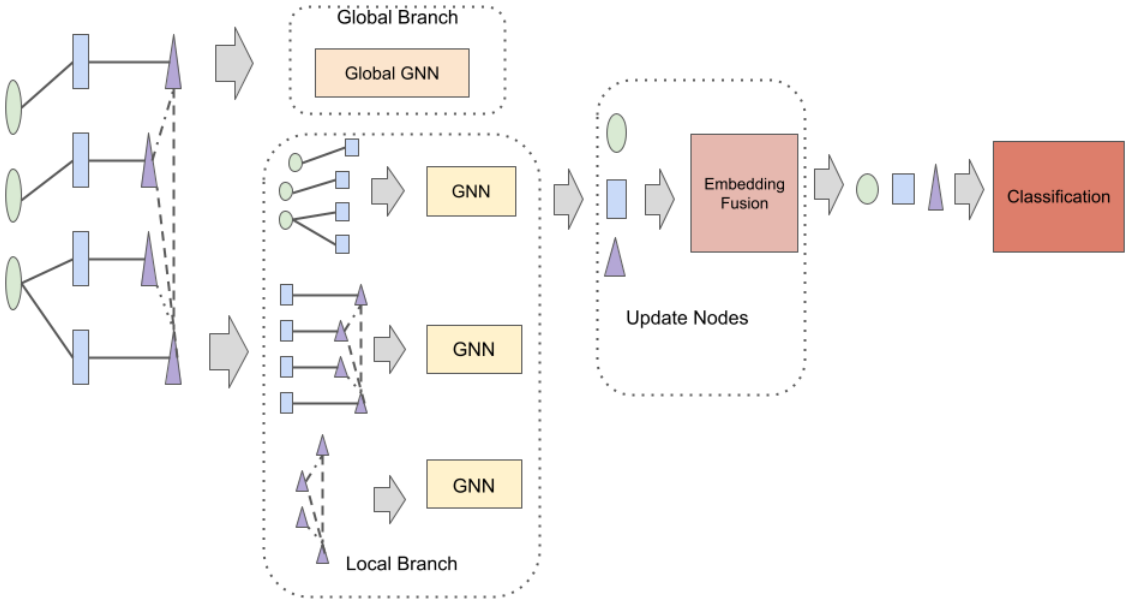

Figure 5.2: The architecture of GL-HGNN.

neural networks, one is the global branch where we employ a GNN block on U-R-P graph to learn nodes representations globally, called global representation. The other is the local branch, we employ GNN blocks on the three sub-graphs to obtain local representations of user, review and product nodes. We use the average operation to aggregate the global and local embedding of each node. Finally, we concatenate the user, review and product nodes embedding together followed by a linear layer for predicting the review helpfulness.

Global GNN Branch. In the U-R-P diagram, if two users purchase the same product or different products that are related, the two user nodes are connected by links between the user node and the review node, and the review node and the product node. Therefore, we can implicitly infer from this graph that the user has similar shopping patterns with other users and the user's past experience in writing reviews. To calculate the global embedding of nodes, we have the following equation.

$$Z_{global}^{U,R,P} = \sigma(GNN(X^{U,R,P}, A^{U,R,P}, W_{global})) \tag{5.2}$$

where GNN denotes the graph neural networks applied on the graph, here we use graph convolutional networks (GCN) for node representations. $X^{U,R,P} \in \mathbb{R}^C$ indicates input embeddings of all nodes in the graph whose feature dimension is C. $A^{U,R,P}$ is the adjacency matrix of the heterogeneous graph. $W_{global} \in \mathbb{R}^{CXH}$ named global matrix is an input-to-hidden weight matrix for a hidden layer with H feature maps used in GNN algorithm. σ is the activation function, we use $softmax$ activation function here. $Z_{global}^{U,R,P} \in \mathbb{R}^H$ is the output matrix representing nodes global representation.

Local GNN Branch. In contrast to the global learning strategy, we focus on subgraphs and learn node representations on different graphs locally. In this case, product nodes are not in the user-review graph and user nodes are not in the review-product graph and product-product graph.

We have the following equations which shows GNN applied on the user-review graph and review-product graph respectively.

$$Z_{local}^{U,R} = \sigma(GNN(X^{U,R}, A^{U,R}, W_{local}^{1})) \tag{5.3}$$

$$Z_{local}^{R,P} = \sigma(GNN(X^{R,P}, A^{R,P}, W_{local}^{2})) \tag{5.4}$$

We have three link types in the product-product graph from the local view, therefore, the adjacency matrix will be different for different link types.

$$Z_{local}^{P,P} = [Z_{local,1}^{P,P}, Z_{local,2}^{P,P}, Z_{local,3}^{P,P}] \tag{5.5}$$

where

$$Z^{P,P}_{local,i} = \sigma(GNN(X^{P,P}, A^{P,P}_i, W^3_{local,i}))$$

After a forward computation, we have new node representations from global and local branches. Then, we need to fuse these global and local representations as well as nodes with multiple feature embeddings under local computation.

$$\begin{cases} X^U = \sigma(Linear(concat(Z^{U,R,P}_{global}, Z^{U,R}_{local}))) \\ X^R = \sigma(Linear(concat(Z^{U,R,P}_{global}, Z^{U,R}_{local}, Z^{R,P}_{local}))) \\ X^P = \sigma(Linear(concat(Z^{U,R,P}_{global}, Z^{R,P}_{local}, Z^{P,P}_{local}))) \end{cases} \tag{5.6}$$

where, a linear block is the following

$$Linear(X) = \Theta X + b$$

To fuse the node embeddings, the user nodes involve $Z^{U,R,P}_{global}$ and $Z^{U,R}_{local}$. We first concatenate the two parts of the updated user node embedding with a linear layer for linear projection, and then we use the $ReLU$ activation function. The process of updating review and product nodes is the same, but involves different node embeddings.

In the final stage, to predict the helpfulness of a particular review, we concatenate the corresponding user, review and product nodes and use a linear layer to predict the helpfulness of the review. The formula can be expressed as follows.

$$y = softmax(Linear(concat(X^u, X^r, X^p))) \tag{5.7}$$

We present the overall algorithm by employing GNNs on heterogeneous graphs to learn global and local node representations. The pseudo-code for learning GL-

Algorithm 1: Algorithm for learning GL-HGNN

Input:U:Set of user nodes, R: Set of review nodes, P: Set of product nodes, $Iter$: Number of iteration, Y: Ground truth of reviews
G = GraphConstruction(U, R, P)
$G^{U,R} \in G$, $G^{R,P} \in G$, $G^{P,P} \in G$
Initialization: $X^{U,R,P} = \{X^U, X^R, X^P\}$
foreach $i \in Iter$ **do**
    $Z^{U,P,R}$ = *GlobalGNN*($\{X^U, X^R, X^P\}$, G)
    $Z^{U,R}$ = *LocalGNN*($\{X^U, X^R\}$, $G^{U,R}$)
    $Z^{R,P}$ = *LocalGNN*($\{X^R, X^P\}$, $G^{R,P}$)
    $Z^{P,P}$ = *LocalGNN*($\{X^P, X^P\}$, $G^{P,P}$)
    X^U = *Fusion*($Z^{U,P,R}$, $Z^{U,R}$)
    X^R = *Fusion*($Z^{U,P,R}$, $Z^{U,R}$, $Z^{R,P}$)
    X^P = *Fusion*($Z^{U,P,R}$, $Z^{R,P}$, $Z^{P,P}$)
    loss = *CrossEntropy*(*cls*(X^U, X^R, X^P), Y)
    Backward propagation to update parameters
end

HGNN is shown as Algorithm 1. We first organize the information from the review text and metadata to get the corresponding user, product and review nodes. We construct the heterogeneous graph G and subgraphs $G^{U,R}$, $G^{R,P}$, $G^{P,P}$ and initialize the node embeddings using the pre-trained ALBERT[28]. GlobalGNN and LocalGNN are applied to the whole graph and subgraphs, respectively, to learn the global and local node representations. We employ a fusion layer to fuse the global and local representations and a linear layer to predict the review helpfulness.

Chapter 6

Experiments

In this chapter, we evaluate our GL-HGNN on eight Amazon public review datasets. Table 3.1 shows the comparison between our proposed method and other machine learning approaches.

Dataset. We first filter those reviews with fewer than 3 voters to exclude noisy data, because it is difficult to determine, even for humans, whether receiving reviews with fewer than 3 voters is helpful. We mark a comment as helpful if the ratio of helpful votes to the total number of votes is greater than 0.75. We partition the dataset in a 7:1:2 ratio based on the timestamps of the comments for training, validation, and test set ranking.

Baselines. To make a fair comparison, we compare our approach with the following baseline from two perspectives: a traditional machine learning model using hand-crafted features and a deep neural network.

(a) The extracted hand-crafted features are listed in Table 2.1. We employ the traditional machine learning models, logistic regression (LR), random forest (RF), decision tree (DT), k-nearest neighbors (KNN), to predict review helpfulness following the existing approaches that rely on features from plain text.

	Clothing Shoes & Jewelry	Grocery & Gourmet Food	Health & Personal Care	Home & Kitchen	Movies TV	Pet Supplies	Tools & Home Improvement	Toys Games
LR	0.625	0.657	0.626	0.621	0.694	0.643	0.634	0.597
RF	0.648	0.671	**0.635**	0.632	**0.719**	0.678	0.647	**0.621**
KNN	0.598	0.646	0.609	0.604	0.705	0.622	0.603	0.566
DT	0.594	0.622	0.588	0.581	0.661	0.606	0.598	0.577
Text-CNN	0.513	0.545	0.584	0.562	0.656	0.576	0.605	0.573
Bi-LSTM	0.582	0.591	0.587	0.590	0.650	0.601	0.603	0.591
RPH-Net	0.605	0.581	0.573	0.583	0.646	0.591	0.596	0.593
AL-BERT	0.591	0.613	0.632	0.620	0.695	0.655	**0.657**	0.613
GL-HGNN (ours)	**0.649**	**0.678**	0.631	**0.648**	0.711	**0.679**	0.652	0.601

Table 6.1: Comparison of GL-HGNN with baselines on AUC. Evaluated on 8 Amazon categories. For all number, higher values are better. *The improvement is statistically significant ($p < 0.05$)*

.

(b) The approach using deep neural networks as the main architecture focuses not only on hand-crafted features from the review text [12, 13], but also considers learning embeddings from the text and incorporating other metadata such as product titles, descriptions, etc [3].

Model Parameter Settings. All models in this experiment use the same text embedding as initialization with dimension 128 from the pre-trained AL-BERT model [28]. In GL-HGNN, we employ graph neural network to learn the node representation, and the hidden dimension of each layer of GCN is 128. For both global and local branches, we use 2 layers of GCN. We use a linear layer for fusion and classification layers.

Experiment environment. We use a Linux server with a 64-bit system (24 core CPU with 3.10GHz, four GPUs RTX 6000 with a memory of 128G). Our algorithm is implemented in Python 3.7.

The datasets are shown in Table 3.1. Since this is a binary classification problem and there is a label imbalance issue as discussed in Chapter 3, we use the Area Under Receiver operating characteristic (AUC) and F1 score as our metrics to evaluate the model performance. Table 6.1 and 6.2 show model the comparison of model performance evaluated by AUC and F1 score, respectively.

	Clothing Shoes & Jewelry	Grocery & Gourmet Food	Health & Personal Care	Home & Kitchen	Movies TV	Pet Supplies	Tools & Home Improvement	Toys Games
LR	0.684	0.701	0.657	0.648	0.733	0.706	0.682	0.577
RF	0.691	**0.716**	0.671	0.678	**0.763**	0.724	0.696	**0.638**
KNN	0.669	0.623	0.584	0.622	0.682	0.647	0.619	0.542
DT	0.669	0.625	0.587	0.631	0.688	0.651	0.635	0.590
Text-CNN	0.626	0.415	0.393	0.517	0.163	0.554	0.507	0.412
Bi-LSTM	0.634	0.438	0.415	0.536	0.219	0.571	0.525	0.431
RPH-Net	0.663	0.458	0.441	0.551	0.289	0.606	0.541	0.454
AL-BERT	0.635	0.597	0.598	0.637	0.678	0.675	0.652	0.566
GL-HGNN (ours)	**0.717**	0.694	**0.683**	**0.682**	0.673	**0.715**	**0.702**	0.604

Table 6.2: Comparison of GL-HGNN with baselines on F1. Evaluated on 8 Amazon categories. For all number, higher values are better. *The improvement is statistically significant ($p < 0.05$)*

.

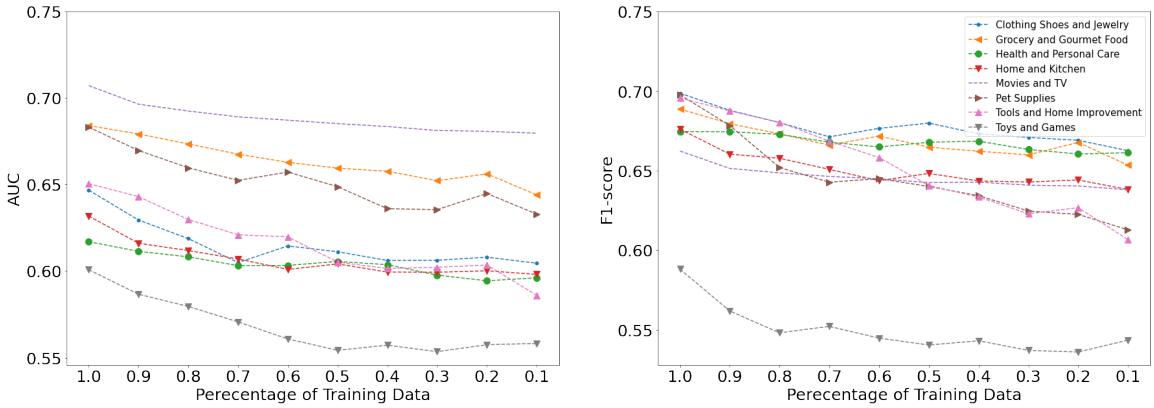

Figure 6.1: Performance of GL-HGCNN trained with different number of training samples. The Y-axis denotes the evaluation metrics and the X-axis denotes the percentage of training samples for training the model.

As shown in Table 6.1 and 6.2, although logistic regression and random forest are naive models, they achieve better performance over the baseline on the classification task. They achieve the best performance in three of the eight categories evaluated by AUC. The state-of-the-art neural networks performed poorly on this task. There could be several reasons, such as the size of the datasets is not large enough and the model cannot learn valuable features for prediction. Our proposed approach GL-HGNN outperforms other baselines on the AUC metric in five out of the eight categories.

We split the dataset according to the temporal information so that samples' dates in training data will not overlap with samples' dates in test data. We then investigate

the robustness of our approach by restricting the number of samples in training data to see the change of model performance on test set.

Figure 6.1 shows the robustness experiment results. We first split the training set into 10 partitions (each partition will have 10% training samples) and keep the test set as the same. We gradually shrink the number of partitions for training, from 100% training set to 10% training set. There is a clear downward trend for both AUC and F1 score when we decrease the number of training samples, but the performance of both two metrics do not drop a lot. For example, for *Toys and Games* category, the F1 score varies from 0.6 to 0.55 when we decrease the number of training data. Performance does not drop by more than 15% in all categories. We believe our approach is robust in this case.

Chapter 7

Conclusion and Future Work

This paper explores the problem of helpfulness prediction for online reviews. Our approach is motivated by (i) the fact that although many standard features are proposed, not all of them are useful and investigating the correlation between features and usefulness is necessary, (ii) the relationship between users, reviews and products should be taken into account, and (iii) we lack sufficient empirical comparisons to show the effectiveness of these methods.

Therefore, we propose the Global-Local Heterogeneous Graph Neural Network (GL-HGNN) to address the above problem. Our contributions are as follows:

(a) We investigate the correlation between features and review helpfulness. We study the relationship between the helpfulness and the five types of features (structural features, lexical features, semantic features and meta data) on eight Amazon datasets.

(b) We propose the GL-HGNN framework, which consists of two parts: one is the construction of a heterogeneous graph, and the other is the use of GNNs on this graph to learn global and local node representations. We use the heterogeneous graph to build connections between users, reviews and products, and then apply

GNNs to this graph to better learn feature representations.

(c) We compare the performance between our approach and other baselines, including traditional machine learning models and deep neural networks. The experimental results show the effectiveness of our approach.

We consider the following future work:

(a) Feedback System of helpfulness improvement. Although we now know which review is helpful, we do not know how to improve the helpfulness of a review if it is not helpful. Users will write better reviews if there is feedback system to help them when they write the reviews.

(b) Knowledge fusion. It is always difficult to predict whether a review of a new product will be helpful or not. By using characteristics of other similar products, we can better predict the helpfulness of reviews.

Bibliography

[1] Wenjing Duan, Bin Gu, and Andrew B. Whinston. The dynamics of online word-of-mouth and product sales—an empirical investigation of the movie industry. *Journal of Retailing*, 84(2):233–242, 2008.

[2] 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.

[3] 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, page 2715–2721, New York, NY, USA, 2019. Association for Computing Machinery.

[4] 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, page 423–430, USA, 2006. Association for Computational Linguistics.

[5] 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, page 423–430. Association for Computational Linguistics, 2006.

[6] 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.

[7] 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.

[8] Pradeep Racherla and Wesley Friske. Perceived 'usefulness' of online consumer reviews: An exploratory investigation across three services categories. *Electronic Commerce Research and Applications*, 11(6):548 – 559, 2012. Information Services in EC.

[9] Lingyun Qiu, Jun Pang, and Kai H. Lim. Effects of conflicting aggregated rating on ewom review credibility and diagnosticity: The moderating role of review valence. *Decision Support Systems*, 54(1):631 – 643, 2012.

[10] Shasha Zhou and Bin Guo. The order effect on online review helpfulness: A social influence perspective. *Decision Support Systems*, 93:77 – 87, 2017.

[11] 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. Solid and Physical Modeling 2012.

[12] 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.

[13] 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, 2018.

[14] Srijan Kumar, Bryan Hooi, Disha Makhija, Mohit Kumar, Christos Faloutsos, and V.S. Subrahmanian. Rev2: Fraudulent user prediction in rating platforms. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, WSDM '18, page 333–341, New York, NY, USA, 2018. Association for Computing Machinery.

[15] Haipeng Chen, Rui Liu, Noseong Park, and V.S. Subrahmanian. *Using Twitter to Predict When Vulnerabilities Will Be Exploited*, page 3143–3152. Association for Computing Machinery, New York, NY, USA, 2019.

[16] 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, page 1–8. Association for Computing Machinery, 2013.

[17] Lionel Martin and Pearl Pu. Prediction of helpful reviews using emotions extraction. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI-14)*, number CONF, 2014.

[18] Hye-Jin Min and Jong C. Park. Identifying helpful reviews based on customer's mentions about experiences. *Expert Systems with Applications*, 39(15):11830 – 11838, 2012.

[19] Kapil Kaushik, Rajhans Mishra, Nripendra P. Rana, and Yogesh K. Dwivedi. Exploring reviews and review sequences on e-commerce platform: A study of helpful reviews on amazon.in. *Journal of Retailing and Consumer Services*, 45:21 – 32, 2018.

[20] 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.

[21] Pei-Yu Chen, Samita Dhanasobhon, and Michael D Smith. All reviews are not created equal: The disaggregate impact of reviews and reviewers at amazon. com. *Com (May 2008)*, 2008.

[22] Raffaele Filieri. What makes online reviews helpful? a diagnosticity-adoption framework to explain informational and normative influences in e-wom. *Journal of Business Research*, 68(6):1261 – 1270, 2015.

[23] Susan M Mudambi and David Schuff. Research note: What makes a helpful online review? a study of customer reviews on amazon. com. *MIS quarterly*, pages 185–200, 2010.

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[25] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[26] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2016.

[27] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda

Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

[28] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. ALBERT: A lite BERT for self-supervised learning of language representations. *CoRR*, abs/1909.11942, 2019.