Decision Support Systems 148 (2021) 113581

Contents lists available at ScienceDirect

Decision Support Systems

journal homepage: www.elsevier.com/locate/dss

Neighbor-aware review helpfulness prediction

Jiahua Du [a,b,*], Jia Rong [c,**], Hua Wang [b], Yanchun Zhang [b]

[a] School of Electronics and Information Engineering, Guangdong Polytechnic Normal University, Guangzhou, Guangdong, China
[b] Institute of Sustainable Industries & Liveable Cities, Victoria University, Melbourne, VIC, Australia
[c] Faculty of Information Technology, Monash University, Clayton, VIC, Australia

ARTICLE INFO

Keywords:
Review helpfulness
Sequential bias
Review neighbors
Context clues
Deep learning

ABSTRACT

Helpfulness prediction techniques have been widely incorporated into online decision support systems to identify high-quality reviews. Most current studies on helpfulness prediction assume that a review's helpfulness only relies on information from itself. In practice, however, consumers hardly process reviews independently because reviews are displayed in sequence; a review is more likely to be affected by its adjacent neighbors in the sequence, which is largely understudied. In this paper, we proposed the first end-to-end neural architecture to capture the missing interaction between reviews and their neighbors. Our model allows for a total of 12 (three selection × four aggregation) schemes that contextualize a review into the context clues learned from its neighbors. We evaluated our model on six domains of real-world online reviews against a series of state-of-the-art baselines. Experimental results confirm the influence of sequential neighbors on reviews and show that our model significantly outperforms the baselines by 1% to 5%. We further revealed how reviews are influenced by their neighbors during helpfulness perception via extensive analysis. The results and findings of our work provide theoretical contributions to the field of review helpfulness prediction and offer insights into practical decision support system design.

1. Introduction

Reading user-generated reviews has become an integral part of online shopping. A recent survey [1] shows that 97% of customers rely on online reviews and 85% of them perceive those reviews as personal recommendations. Nonetheless, online reviews are uneven in quality: when a product accumulates reviews, distinguishing high-quality ones from those of random quality can be difficult. To combat information overload, many e-commerce platforms have asked users to vote whether a review is helpful. Decision support systems are then developed to rank reviews based on the helpfulness votes. Although efficient, these systems are limited by scarce voting data, and thus require automatic approaches to facilitate locating helpful reviews.

Helpfulness prediction aims to identify high-quality reviews. Prior literature has explored various features and models [2,3,36], but largely assumed that customers are unbiased, perceiving a review's helpfulness only based on information from itself. In practice, customers' perception on a review can be affected by other reviews they have read. We demonstrate this idea with two examples in Table 1. Suppose Emily is reading reviews of a dating website and Review X is the first one she reads, determining its helpfulness would be difficult due to insufficient information. If Emily had read the three reviews with similar opinions, she might have been more likely to agree on Review X. Likewise, Emily might have been more likely to vote against Review Y due to the contrasting opinions. Both examples show that customers can be affected by context clues – information learned beyond a review – when perceiving helpfulness. The challenge is thus how to utilize context clues to improve helpfulness prediction.

In this work, we hypothesize that a review's helpfulness depends on itself and the interaction with its context clues. We focus on context clues learned from a review's adjacent counterparts (henceforth called neighbors) in the review sequence. The intuition is that customers only have limited patience for a few reviews [4] and a review's neighbors are more likely to be seen and thus influence the review. We propose a deep neural architecture for Neighbor-Aware helpfulness Prediction (NAP) and investigate how a review will be influenced by its neighbors during helpfulness perception.

The contributions of our work are threefold. (1) Our work is one of

* Principal corresponding author.
** Corresponding author.
E-mail addresses: jiahua.du.edu@gmail.com (J. Du), jiarong@acm.org (J. Rong), hua.wang@vu.edu.au (H. Wang), yanchun.zhang@vu.edu.au (Y. Zhang).

https://doi.org/10.1016/j.dss.2021.113581
Received 9 June 2020; Received in revised form 27 April 2021; Accepted 27 April 2021
Available online 1 May 2021

Table 1
Customers can perceive the same review differently if they have read other reviews.

Review X: 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.
If customers had read the following reviews before Review X:
1. I *met an awesome man*. 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. 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*.
Review Y: Senior dating? I signed up my dad 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).
If customers had read the following reviews before Review Y:
1. I was disappointed and they took my money but *no dates after 6 months* of subscriptions. No one sent me any email or responded, 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.

the first studies that consider the interaction between a review and its neighbors. In particular, NAP is the first end-to-end neighbor-aware framework that learns context clues of a review directly from review texts. (2) We further design three schemes for selecting neighbors of a review and four aggregation schemes to construct context clues from the selected neighbors. (3) Our findings reveal how reviews are influenced by their neighbors and offer insights for future researchers and business practitioners.

The rest of this paper is organized as follows. Section 2 surveys relevant literature. Section 3 defines the task of neighbor-aware helpfulness prediction and presents the NAP architecture. Section 4 describes experimental design and Section 5 analyzes the results obtained. Section 6 discusses findings, implications, and limitations. Section 7 concludes this work and presents future directions.

2. Related work

This section identifies bias inherent in review helpfulness when users process a sequence of reviews and then surveys recent studies using review neighbors to alleviate sequential bias during helpfulness prediction.

2.1. Sequential Bias in helpfulness perception

Customers are inevitably affected by social influence [5,6] when perceiving review helpfulness: since reviews are sequentially displayed, how a review is positioned and presented [7–9] can result in different perceived helpfulness. Experimental studies [10–13] found that reviews that customers have read can bias those they are about to. One explanation [14] is that customers learn a relative majority opinion (i.e., initial beliefs about a product) from earlier reviews and compare the majority opinion with later reviews serving as new opinions about a product. The consistency [15] (in terms of text informativeness, valence, etc.) between the two types of opinions [16–18] thus affect customers' helpfulness perception. Another explanation [19] is that helpfulness voting is used to "correct" reviews that customers believe should have a lower/higher ranking in the sequence. Consequently, helpfulness evaluation rarely occurs independently in practice.

2.2. Neighbor-aware helpfulness prediction

To assess the impact of sequential bias, recent studies have attempted to learn context clues for a review from other reviews during helpfulness prediction. Zhou et al. [20] used review orders as context clues and recorded the position of individual reviews sorted by timestamps in a sequence; Jorge et al. [21] referred to context clues as the incremental information entropy of each review: the number of words in a review that neither have been mentioned in its preceding reviews nor the product description. Hong et al. [22] measured the sentiment difference of a review from the mainstream opinion of all reviews. Similarly, Lu et al. [23] computed the divergence between a review's opinion and the overall opinion of all reviews via a unigram language model. These studies have two main drawbacks. First, some of the context clues were based on review orders at the time of collection. Since many platforms adopt review ranking mechanisms, one snapshot of reviews cannot reflect the ranking dynamics, and thus these studies have yet to model the true order information [24]. Second, customers were assumed to be aware of the whole sequence of reviews when determining a review's helpfulness, but most customers only have a low degree of patience [4] when reading online reviews.

Our work aims to extend existing studies and overcome the aforementioned drawbacks. (1) We create a novel dataset containing six domains of online reviews for experimentation. This dataset is advantageous since reviews are collected from platforms that maintain static review orders over time. (2) Instead of building shallow peripheral cues [25] (e.g., a review's star rating, increased number of unique words, or detected sentiment), we employ deep learning techniques to learn context clues with richer semantics from review texts. (3) We adopt local neighbors in place of the whole review sequence and construct context clues in a more flexible and comprehensive manner.

3. Methodology

We formulate helpfulness prediction as a binary text classification task. Let $\mathbf{S} = (S_i)_{i=1}^N$ be a list of reviews and $\mathbf{y} = (y_i)_{i=1}^N$ the corresponding helpfulness labels, where $y = 1$ is helpful and $y = 0$ unhelpful. Given model parameters θ, most studies predict the helpfulness of a reivew S_i using only information from the review $P\left(\widehat{y}_i|S_i;\theta\right)$; we henceforth call this type of methods *independent* helpfulness prediction. This work instead contextualizes each review S_i within its adjacent counterparts $\mathbf{T}_i \subset \{\mathbf{S}\backslash S_i\}$ in the review sequence $\mathbf{S}$; $\widehat{y}_i$ is computed by considering information from its neighbors $P\left(\widehat{y}_i|S_i,\mathbf{T_i};\boldsymbol{\theta}\right)$, and thus *neighbor-aware* helpfulness prediction.

We present NAP, an end-to-end deep neural architecture for neighbor-aware helpfulness prediction. As illustrated in Fig. 1, NAP first encodes each review S into a representation h and learns the context clues c of a review from its associated neighbors; the neighbor-aware representation $\widehat{\mathbf{h}}$ of S results from the interaction between h and c, which is used to predict helpfulness.

3.1. Review text encoding

Let each review $S = (x_i)_{i=1}^n$ be a sequence of n words and V the vocabulary that indexes all unique words in $\mathbf{S}$. Given a lookup table $\mathbf{E} \in \mathbb{R}^{|V|\times d}$, each word $x \in V$ is associated with a d-dimensional vector (i.e., embedding) $\mathbf{e}_x = \mathbf{E}^T\mathbf{x}$, where $\mathbf{x} \in \mathbb{R}^{|V|}$ is the one-hot encoding of x. As a result, S can be represented by an embedding matrix $\mathbf{X} = [\mathbf{e}_{x_1}, \mathbf{e}_{x_2}, \ldots, \mathbf{e}_{x_n}] \in \mathbb{R}^{n\times d}$.

NAP encodes review semantics using a Convolutional Neural Network (CNN) [26] of m kernels $\{W_c \in \mathbb{R}^{l\times d\times m}, b_c \in \mathbb{R}^m\}$. Each kernel is a sliding window of l words over $\mathbf{X}$. The feature maps $\mathbf{H}$ are then activated using Exponential Linear Units (ELUs) [27], followed by column-wise max pooling [28] that selects the most salient features of a review. The final representation $\mathbf{h} \in \mathbb{R}^m$ of a review is computed as follows:

$$\mathbf{H} = \mathrm{ELU}(\mathbf{X}^*\mathbf{W}_c + \mathbf{b}_c), \tag{1}$$

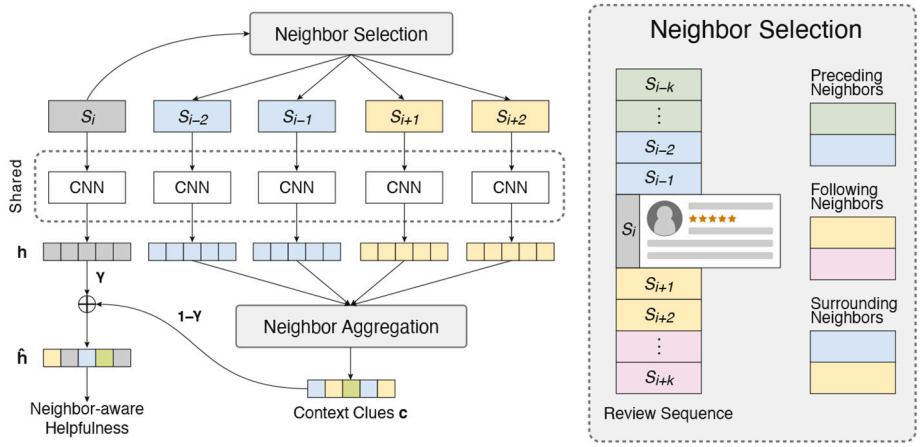

Fig. 1. The NAP architecture. As an example, four surrounding neighbors are selected to learn the context clues of review S_i.

$$\mathbf{h} = max(\mathbf{H}) \tag{2}$$

We currently aim to investigate how a review's helpfulness will be influenced by its neighbors instead of complex model construction. For sophisticated text representations, one can implement more advanced frameworks [29,30] to learn embeddings of higher granularity and adaptive embeddings.

3.2. Context clue construction

The context clues of the i-th review $S_i \in \mathbf{S}$ are learned from its neighbors $\{S_j | j \in [i-k, i+k], j \neq i, k \in \mathbb{N}^+\}$. We explore three neighbor selection schemes for context clue construction:

$$\mathbf{T}_i = \begin{cases} (S_j)_{j=i-1}^{i-k}, & k \text{ preceding neighbors,} \\ (S_j)_{j=i+1}^{i+k}, & k \text{ following neighbors,} \\ (S_j)_{j=i-\lceil k/2 \rceil}^{i+\lceil k/2 \rceil} \backslash S_i, & 2\lceil k/2 \rceil \text{ surrounding neighbors.} \end{cases} \tag{3}$$

The selected neighbors $\mathbf{T}_i$ are regarded as reviews a customer has read prior to S_i. NAP accepts both preceding and following reviews because the review order in $\mathbf{S}$ does not necessarily reflect the reading order [24]: customers could have read reviews displayed above or below S_i when voting its helpfulness.

Given a review S, the embeddings of its K neighbors $\mathbf{T}$ are stacked into a matrix $\mathbf{C} \in \mathbb{R}^{K*m}$. The context clues of S, denoted by $\mathbf{c} \in \mathbb{R}^m$, result from a neighbor aggregation function $f : \mathbb{R}^{K*m} \rightarrow \mathbb{R}^m$ that merges the K neighbor embeddings, $\mathbf{c} = f(\mathbf{C})$; f is set as an identity map when $K = 1$. The context clues imitate customers forming their first impression of a product from reviews they have read before S. NAP introduces four schemes for f:

- **Average (AVG)** The first scheme borrows the idea from the neural the bag-of-words model [31] where a sentence embedding is represented by the centroid of its word counterparts. This model has been used in many natural language processing tasks, including review text modeling [32,33]. In this work, context clues (analogous to a sentence) are represented by bag-of-neighbors (analogous to words) model. The identical weights show equal importance of neighbors when context clues.

$$\mathbf{c} = \frac{1}{K} \sum_{i=1}^{K} \mathbf{C}_i. \tag{4}$$

- **Weighted Average (WAVG)** The second scheme replaces the fixed weights in Eq. (4) with learnable parameters. This allows neighbors to have different importance when forming context clues. We learn the parameters via an attention mechanism [34], which employs a query vector $\mathbf{u}_a \in \mathbb{R}^m$ as the learnable function. Context clues result from the weighted average of the K neighbor embeddings.

$$z_i = tanh(\mathbf{u}_a^\top \mathbf{C}_i), \tag{5}$$

$$\alpha_i = \frac{exp(z_i)}{\sum_{j=1}^{K} exp(z_j)}, \tag{6}$$

$$\mathbf{c} = \sum_{i=1}^{K} \alpha_i \mathbf{C}_i. \tag{7}$$

- **Feature Regression (FR)** The third scheme merges neighbor embeddings on a dimension level. Each dimension of a review embedding suggests a certain type of latent characteristic, which may attract different reading interests. We compute the weights for $\mathbf{C}$ using a similar attention mechanism, followed by column-wise softmax:

$$\mathbf{Z} = tanh(\mathbf{W}_b \otimes \mathbf{C}), \tag{8}$$

$$\beta_{ij} = \frac{exp(\mathbf{Z}_{ij})}{\sum_{k=1}^{K} exp(\mathbf{Z}_{kj})}, \tag{9}$$

$$c_j = \sum_{k=1}^{K} \beta_{kj} \mathbf{C}_{kj}, \tag{10}$$

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

- **Spatial Feature Regression (SFR)** The fourth scheme extends FR by interacting with neighbors when learning context clues. Neighbors closer to the target review tend to attract more attention. Since neighbors read earlier may also affect those later, we share information of closer neighbors with farther ones such that:

$$\widehat{\mathbf{C}}_i = \begin{cases} \sum_{k=i}^{K} \mathbf{C}_k, & \text{Preceding neighbors,} \\ \sum_{k=1}^{i} \mathbf{C}_k, & \text{Following neighbors.} \end{cases} \tag{11}$$

Surrounding neighbors can be thought of as a special case where preceding and following neighbors coexist. The enhanced matrix $\widehat{\mathbf{C}}$ is then passed to Eqs. (8)–(10) in place of $\mathbf{C}$.

3.3. Neighbor-aware helpfulness prediction

Finally, NAP contextualizes a review within its neighbors by aggregating the representation of the review $\mathbf{h}$ and its context clues $\mathbf{c}$ via linear combination:

$$\widehat{\mathbf{h}} = \gamma \mathbf{h} + (1-\gamma)\mathbf{c}. \tag{12}$$

In the aggregation, $\mathbf{c}$ can be thought of as a user's initial belief (the relative majority opinion) of a product, whereas $\mathbf{h}$ a new opinion to be compared with. The aggregation factor $\gamma \in [0,1]$ controls how much a review is influenced by its neighbors. When $\gamma = 1$, a review's helpfulness relies on the review per se; when $\gamma = 0$, the helpfulness relies exclusively on its context clues.

The neighbor-aware representation of a review $\widehat{\mathbf{h}}$ is forwarded into a logistic regression layer to predict its helpfulness $\widehat{y}$:

$$\widehat{y} = \sigma\left(\mathbf{W}_o^\top \widehat{\mathbf{h}} + b_o\right) \tag{13}$$

NAP is trained via cross entropy minimization over M samples:

$$\mathscr{L} = -\frac{1}{M}[\mathbf{y}^\top log(\widehat{\mathbf{y}}) + (1-\mathbf{y})^\top log(1-\widehat{\mathbf{y}})] + \frac{\lambda}{2}\|\mathbf{W}_c\|^2. \tag{14}$$

where $\widehat{\mathbf{y}}$ are the predicted helpfulness of the M samples and $\mathbf{y}$ ground truth labels. The weight decay λ used for CNN filter regularization is added to reduce overfitting.

4. Experimental design

We conducted extensive experiments to benchmark NAP against a series of baselines. We first describe the datasets used throughout the experiments in Section 4.1, and then list the baselines in Section 4.2 for comparison. Finally, we present hyperparameters for training NAP and the baseline models.

4.1. Data description

NAP requires review-neighbors pairs $(S, \mathbf{T})$ to validate neighbor-aware helpfulness prediction. Most existing studies collected data from platforms that adopt review ranking mechanisms, which inevitably brings early bird bias [35] into reviews. Additionally, reviews collected at one snapshot do not reveal their previous orders. These data thus failed to fit our methodology. In this work, we leveraged data from SiteJabber[1] and ConsumerAffairs[2] wherein reviews were displayed to customers in reverse chronological order. Both platforms maintained static review orders over time: early reviews will be pushed back in the sequence and cease gaining advantages once an item accumulates more reviews. As such, one snapshot of these data suffice for training NAP. It is worth noting that NAP can also be applied to reviews with ranking dynamics if given multi-snapshot data.

We collected a total of 169,126 reviews posted before April 29, 2019 from SiteJabber and ConsumerAffairs, along with the review metadata. We chose three domains from each platform that had the largest number of reviews to ensure sufficient data fitting deep learning models. Each domain contains a list of items (products or companies) and each item a sequence of ordered reviews. For simplicity, the six domains are called D1, D2, and so on. Table 2 shows the descriptive statistics. As the first step to investigate the relationship between a review and its sequential neighbors, we focused on encoding review texts, the most common and platform-independent element [36], into helpfulness.

We pre-processed the collected reviews to improve data quality. (1) Reviews that had longer exposure were more likely to receive votes [35]. To avoid early bird bias, we removed reviews posted in early months in which an item accumulated less than 15 (at least one every other day) reviews. (2) Similarly, we removed recent reviews due to insufficient exposure time for voting. (3) To ensure each item has sufficient reviews to build training samples after (1) and (2), we only considered items with a minimum of 100 remaining reviews based on our experience. (4) Each review was lowercased and tokenized, followed by the removal of articles "a", "an", and "the". (5) We kept the most frequent 30,000 terms as vocabulary [36] to reduce the execution cost of model training. Numeric values in reviews were replaced by <NUM> and names of items <ORG>. Reviews with few votes [37,38] were usually filtered out to learn more robust models; we skipped this because review order is important for training NAP.

Following [32], we labeled a review as helpful if it received at least two votes and unhelpful otherwise. For each item, we partitioned reviews into training, validation, and test sets using the 80%-10%-10% splits [36]; we applied chronological split [23] to preserve review order. We replaced out-of-vocabulary words (terms that exist in the training set but do not in the validation/test set) with <UNK>. For each domain, review-neighbors pairs $(S, \mathbf{T})$ were assembled within each partition (Eq. (3)) and gathered across items. To avoid class imbalance, helpful review-neighbors pairs are randomly sampled to have the same number as unhelpful ones and vice versa.

4.2. Baselines

We first evaluated NAP on independent and neighbor-aware helpfulness prediction. Besides the CNN encoder defined in Eq. (1) and (2), we adopted Multilayer Perceptron (MLP) to encode review texts. Specifically, the embedding matrix $\mathbf{X}$ is transformed via average pooling, followed by two fully-connected layers. We denoted the two types of independent prediction as $\mathbf{I}$ and $\mathbf{I}_{\text{MLP}}$, and compared them with their neighbor-aware counterparts.

We further benchmarked NAP against six state-of-the-art contextualized methods. For a fair comparison, we predicted a review's helpfulness by concatenating its context clues extracted as per each baseline and its text representation learned by the CNN encoder in $\mathbf{I}$.

- $\mathbf{I} + \mathbf{ORD}$: The first three baselines operationalize review orders. $\mathbf{I} + \mathbf{ORD}_{\text{D}}$ sorts reviews by their timestamps [20]. Let R be reviews sorted from latest to oldest and d_r the posted date of a review $r \in R$. Reviews $R_{d'} \equiv \{r | d_r = d'\}$ posted on the same day $d' \in \{d_r | r \in R\}$ share the same order $[\sum_{d<d'} N(R_d) + 1]^{-1}$, where $N(R_d)$ is the cardinality of R_d. Similarly, $\mathbf{I} + \mathbf{ORD}_{\text{R}}$ sorts reviews based on their star ratings and $\mathbf{I} + \mathbf{ORD}_{\text{V}}$ the number of helpful votes in descending order.
- $\mathbf{I} + \mathbf{CON}$: This baseline measures the conformity [23] of a review to the others of an item. Each review $r \in R$ is vectorized into its TFIDF representation $\mathbf{u}_r$. The conformity calculates the Kullback–Leibler divergence between a review $\mathbf{u}_r$ and the overall opinion $\overline{\mathbf{u}} = \frac{1}{|R|}\sum_{r \in R} \mathbf{u}_r$.
- $\mathbf{I} + \mathbf{POL}$: This baseline measures the sentiment divergence [22] of a review from the others of an item. Each review $r \in R$ is categorized into $c_r \in \{\text{negative}, \text{neutral}, \text{positive}\}$ based on the ratio of positive and negative words $p_r \in [-1, 1]$ in the review. The divergence computes $|p_r - \overline{p}|$, where $\overline{p} = \frac{1}{|R'|}\sum_{r' \in R'} p_{r'}$ is the mainstream opinion, $R' \equiv \{r' | c_r = c'\}$, and $c' \in c_r$ the majority category in R.
- $\mathbf{I} + \mathbf{ENT}$: This baseline measures the incremental entropy [21] of reviews of an item. Let R_n be the n-th review, vocab(R_n) the number of unique words occurred in R_1, R_2, …, R_n. The incremental entropy computes the increased number of unique words vocab(R_n) − vocab(R_{n-1}).

4.3. Hyperparameters

Table 3 lists the main hyperparameters used for training NAP and the baselines. The lookup table $\mathbf{E}$ associates each word with a vector representing the word's semantic meaning. In practice, $\mathbf{E}$ can be initialized with random values or pre-trained embeddings. Pre-trained embeddings were learned based on the distributional hypothesis [39] that words occurring in the same contexts tend to have similar meanings. As a result, words with similar semantics are close to each other in the learned vector space. Following [40], we chose the pre-trained GloVe embeddings [41] for word vector initialization. These vectors were

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

Table 2
Descriptive statistics of the balanced doamins after pre-processing.

Domain		#Reviews	#Words	$\frac{\#Words}{\#Reviews}$	#Sentences	$\frac{\#Sentences}{\#Reviews}$	$\frac{\#Words}{\#Sentences}$
D1	Dating	4054	359,369	88.65	27,035	6.67	12.91
D2	Wedding Dresses	5294	456,602	86.25	36,909	6.97	12.67
D3	Marketplace	6964	581,456	83.49	46,222	6.64	12.31
D4	Car Insurance	2932	398,341	135.86	27,004	9.21	14.42
D5	Travel Agencies	8156	1,168,941	143.32	78,408	9.61	14.67
D6	Mortgages	4602	652,223	141.73	44,955	9.77	14.13

Table 3
NAP hyperparameters.

Hyperparameters		Values
Review Text Encoding	Lookup table **E**	300-d static GloVe vectors
	Number of kernels m	100
	Kernel patch size l	3
	Weight decay λ	5×10^{-4}
Context Clue Construction	Number of neighbors K	[1, 10]
	Aggregation factor γ	0.5
Training	Optimizer	Adam [44]
	Batch size	64
	Early stopping	10 epochs

trained on text extracts of 840 billion tokens and have been successfully adopted in many research areas. Compared with random values, pre-trained embeddings can better encode word semantics inherent in review texts, particularly at the start of model training.

The remaining weights in NAP were set by the Glorot uniform initializer [42]. The range of K is inspired by a consumer purchase behavior study [43] where it stated that the majority of consumers read between 1 and 10 reviews before making a purchase. We fixed the random seed for reproducibility and trained every model five times to ensure model robustness.

5. Results and analysis

We reported experimental results and demonstrated the effectiveness of NAP. Section 5.1 first shows that NAP outperforms the baselines, followed by ablation studies in Section 5.2 confirming that the success of NAP lies in interacting a review with its neighbors. Section 5.3 investigates how NAP's main neighbor settings affect its prediction and measured the trade-off between model accuracy and complexity. Finally, Section 5.4 showcases that NAP learns better review representations for helpfulness prediction. Throughout this work, model performance is measured by classification accuracy.

5.1. Comparison with baselines

Table 4 reported the accuracy of NAP by neighbor type. For simplicity, we denoted NAP using preceding, following, and surrounding neighbors as $\mathbf{I}+\mathbf{P}$, $\mathbf{I}+\mathbf{F}$, and $\mathbf{I}+\mathbf{S}$, respectively; we also denoted $\mathbf{I}+\mathbf{P}_{\text{MLP}}$, $\mathbf{I}+\mathbf{F}_{\text{MLP}}$, and $\mathbf{I}+\mathbf{S}_{\text{MLP}}$ as the neighbor-aware counterparts of $\mathbf{I}_{\text{MLP}}$. NAP results that beat all the baselines are in italic and those of the highest accuracy in bold. We further conducted t-tests [45] between (1) the CNN and MLP encoder, (2) NAP and the contextualized baselines, and (3) neighbor selection schemes used in NAP. Due to space limitations, we only reported the results of (2) in Table 5; the results of (1) and (3), as will be discussed, were consistent across domains and thus omitted.

The first two blocks of Table 4 compared the CNN encoder with the MLP encoder on both independent and neighbor-aware prediction. As shown, both encoders showed a similar trend in performance and confirmed that including context clues improved prediction accuracy. The accuracy scores of $\mathbf{I}_{\text{MLP}}$, $\mathbf{I}+\mathbf{P}_{\text{MLP}}$, $\mathbf{I}+\mathbf{F}_{\text{MLP}}$, and $\mathbf{I}+\mathbf{S}_{\text{MLP}}$ were all significantly lower than their CNN counterparts except for $\mathbf{I}_{\text{MLP}}$ on D3.

Table 4
NAP against the baselines. The aggregation scheme and number of neighbors that produce the highest accuracy are listed below.

	D1	D2	D3	D4	D5	D6
$\mathbf{I}_{\text{MLP}}$	80.05	63.40	81.20	68.77	63.98	68.26
$\mathbf{I}_{\text{MLP}}+\mathbf{P}$	79.62	65.78	82.12	71.23	63.17	67.39
$\mathbf{I}_{\text{MLP}}+\mathbf{F}$	81.96	63.05	81.66	73.93	64.56	68.78
$\mathbf{I}_{\text{MLP}}+\mathbf{S}$	81.34	65.55	82.17	73.93	63.33	68.96
I	86.27	70.04	81.63	72.21	67.15	69.39
I + **P**	*89.90* FR/9	*70.98* FR/10	*83.56* AVG/10	*74.84* SFR/6	*67.96* FR/10	**70.87** WAVG/10
I + **F**	*90.86* SFR/10	*71.17* AVG/7	**83.83** FR/10	*74.59* WAVG/3	**68.41** FR/7	*70.43* WAVG/6
I + **S**	**90.91** WAVG/8	**71.25** WAVG/10	*83.80* FR/10	**75.00** FR/6	*67.80* WAVG/8	*70.35* WAVG/4
$I + ORD_{\text{D}}$	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

CNN is expected to be a more suitable text encoder because it was capable of locating most salient words and phrases in a review, whereas MLP regarded words equally.

The last two blocks of Table 4 compared NAP against six contextualized methods for helpfulness prediction. Overall, NAP significantly outperformed all the baselines by 1% to 5% across domains, with only two exceptions: $\mathbf{I}+\mathbf{ORD}_{\text{R}}$ on D2 and $\mathbf{I}+\mathbf{ORD}_{\text{V}}$ on D5. The promising results of NAP show the efficacy of learning context clues directly from review texts, compared with the baselines that either used external review metadata or peripheral cues of review texts. In contrast, the context clues learned by the six baselines yield less than 1% accuracy gains across domains and are less robust: most improvements are observed on D1 and D2; the context clues influence little on **I** or even diminished the performance on D3, D4, and D6.

5.2. Ablation studies

The accuracy gains of NAP can result from three reasons: (i) the interaction between a review S and its neighbors **T**, (ii) the exclusive context clues learned from **T**, or (iii) increase of review data for model training. We aimed to discover the reason by considering the following NAP variants:

- **P/F/S**: We set $\gamma = 0$ in Eq. (12) to perform neighbor-only prediction that used merely context clues c for helpfulness modeling. The three types of neighbors are all considered.
- **I + R**: Neighbor-aware prediction where context clues **c** are learned from K reviews randomly selected from the same domain.
- **I + N**: Neighbor-aware prediction where context clues **c** draw

Table 5
Significance tests between NAP and the contextualized baselines.

		$\mathbf{ORD}_D$	$\mathbf{ORD}_R$	$\mathbf{ORD}_V$	**I + CON**	**I + POL**	**I + ENT**	I
D1	I + P	0.000***	0.000***	0.000***	0.000***	0.000***	0.000***	0.000***
	I + F	0.000***	0.000***	0.000***	0.000***	0.000***	0.000***	0.000***
	I + S	0.000***	0.000***	0.000***	0.000***	0.000***	0.000***	0.000***
D2	I + P	0.066*	0.848	0.058*	0.224	0.43	0.404	0.046**
	I + F	0.009***	0.416	0.008***	0.052*	0.125	0.096*	0.008***
	I + S	0.003***	0.253	0.003***	0.022**	0.055*	0.034**	0.003***
D3	I + P	0.000***	0.000***	0.000***	0.000***	0.000***	0.000***	0.001***
	I + F	0.000***	0.001***	0.000***	0.000***	0.000***	0.000***	0.002***
	I + S	0.000***	0.000***	0.000***	0.000***	0.000***	0.000***	0.000***
D4	I + P	0.000***	0.000***	0.002***	0.000***	0.008***	0.000***	0.000***
	I + F	0.000***	0.000***	0.003***	0.000***	0.012**	0.000***	0.000***
	I + S	0.000***	0.000***	0.001***	0.000***	0.006***	0.000***	0.000***
D5	I + P	0.069*	0.115	0.57	0.078*	0.384	0.122	0.070*
	I + F	0.001***	0.011**	0.184	0.001***	0.031**	0.020**	0.001***
	I + S	0.236	0.264	0.768	0.269	0.664	0.242	0.248
D6	I + P	0.004***	0.006***	0.011**	0.005***	0.010***	0.006***	0.008***
	I + F	0.006***	0.005***	0.018**	0.004***	0.017**	0.008***	0.012**
	I + S	0.004***	0.002***	0.014**	0.001***	0.016**	0.004***	0.008***

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

random values from a uniform distribution within the range $[0,1]$. This variant can also be thought of as introducing noise into **I**.

5.2.1. Neighbor-aware versus neighbor-only prediction

We first validated the role of neighbors by comparing **I + P**, **I + F**, and **I + S** respectively with **P**, **F**, and **S**. As depicted in Fig. 2, neighbor-only methods received significantly lower accuracy than their neighbor-aware counterparts across domains. The only exception ($K = 7$ following neighbors weighted using WAVG on D6) improved less than 0.1%. Overall, both the neighbor-aware and neighbor-only methods benefit from neighbors, which can be treated as prior knowledge. Involving more neighbors thus helps **P**, **F**, and **S** accumulate context clues, including those that might have been mentioned in the targeted review. This explains why the accuracy of **P**, **F**, and **S** is gaining faster as K increases and less likely to plateau. Still, those accumulated clues hardly cover all information contained in the targeted review; as a result, the neighbor-aware methods can achieve higher accuracy than the neighbor-only counterparts with far fewer neighbors. Without knowing the targeted review, **P**, **F**, and **S** are less stable across aggregation schemes and neighbor types.

In some cases, the neighbor-only methods show comparable predictive power to independent helpfulness prediction. For instance, the accuracy of **P**, **F**, and **S** on D1 is close to **I** at $K = 10$; on D4, neighbor-only methods outperform **I** using $K \geq 4$ reviews. This suggests that the context clues of a review can sometimes approximate its helpfulness. In most cases, however, the neighbor-only methods are ineffective for helpfulness prediction. The results above strongly evidenced that NAP relies on the interaction between a review and its neighbors.

5.2.2. Context clues learned from neighbors versus non-neighbors

We then validated whether the effectiveness of NAP simply results from the increase of data. As shown in Fig. 3, NAP received lower accuracy across domains when using non-neighbors (**I + N** and **I + R**) than neighbors (**I + P**, **I + F** and **I + S**) and even no neighbors **I**. Although involving more random reviews tends to improve **I + R**, the accuracy is worse than if not comparable to **I + N** across domains. This suggests that context clues learned from random reviews harm neighbor-aware prediction even more than uniform noise. When using context clues only for prediction, **N** acts similarly to random guessing and **R** fluctuates around **N** regardless of K. Compared with **P**, **F**, and **S**, accumulating random information cannot form effective context clues. Thus, NAP relies on learning a review's specific context clues from its neighbors and simply including arbitrary reviews does not lead to essential improvement.

5.3. Sensitivity analysis on neighbor settings

5.3.1. Number of neighbors

Overall, NAP improved as the number of neighbors increases and then plateaued. As illustrated in Fig. 2, NAP was initially inferior to **I** because context clues were learned from few neighbors that possess insufficient information; this analogizes **I + N** with another form of noise. NAP started to improve and outperform **I** when context clues encoded more neighbors. In our experiments, NAP beat **I** within the first five neighbors and all domains but D5 achieved so within only the first two neighbors. Once information needed for contextualization saturated, NAP improved little even involving extra neighbors. In our experiments, NAP engaged on average eight neighbors for context clue construction but the accuracy gains majorly occurred in the first few neighbors and were within 1.5% after that. This confirms that neighbors closer to a review drive the bulk of the influence [46] on helpfulness perception.

5.3.2. Neighbor selection and aggregation schemes

We first compared the three neighbor selection schemes used in NAP. We selected **I + P** as the baseline and compared its performance with that of **I + F** and **I + S**. As shown in Fig. 4, the accuracy difference between the neighbor types is mostly within 2%. Further significance tests revealed that although review-neighbors interactions had been proven effective, most of the selection schemes did not significantly outperform one another. This suggests customers may select a review's surrounding neighbors in a more flexible manner.

In a similar vein, we compared the four neighbor aggregation schemes. As shown in Fig. 5, WAVG and FR were more frequently used than AVG and SFR. AVG is robust for context clue construction, with less than 1% difference in accuracy in most cases and 2% in all cases. The highest accuracy scores across domains were majorly achieved by either WAVG or FR, necessitating finer-grained schemes to learn context clues from neighbors of uneven quality. Whereas SFR usually received lower accuracy than other schemes. This suggests further analysis on the interaction mechanism among neighbors.

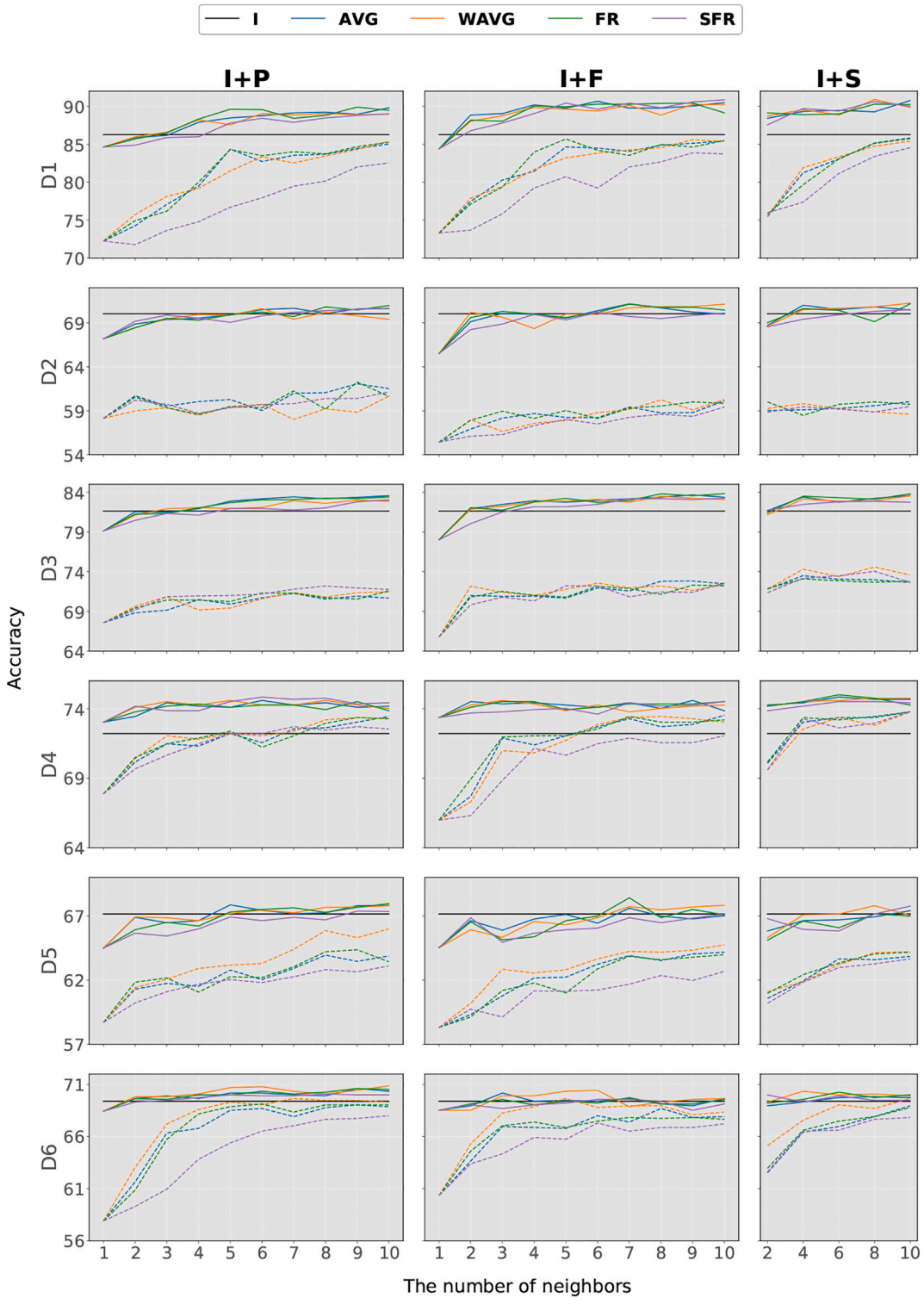

Fig. 2. NAP on different neighbor settings. Dotted lines are the neighbor-only counterparts of the neighbor-aware methods.

5.3.3. Aggregation factor

Fig. 6 analyzes the aggregation factor γ. We used the neighbor settings in Table 4 and varied γ from 0.1 to 0.9 incremented by 0.1; the two cases $\gamma = 0$ (neighbor-only prediction) and $\gamma = 1$ (independent prediction) have been reported in previous sections. Overall, NAP peaked at around $\gamma = 0.5$ and decreased as γ moved towards 0 and 1. This shows the effectiveness of review-neighbors interactions and suggests that neither excessive dependence on a review nor its neighbors facilitated helpfulness prediction. Besides, NAP is more sensitive to context clues in $\gamma \in [0.1,0.5]$ than in $\gamma \in [0.5,0.9]$: D2 and D3 showed high sensitivity, followed by D1 and D5; D4 and D6 are relatively less sensitive to γ. One explanation is domain-specific characteristic difference such as the homogeneity of review opinions towards an item.

5.3.4. Trade-off between performance and complexity

Table 6 summarizes the complexity of NAP during context clue construction. As discussed, using an excessive number of neighbors and/or overcomplicated aggregation schemes does not guarantee a

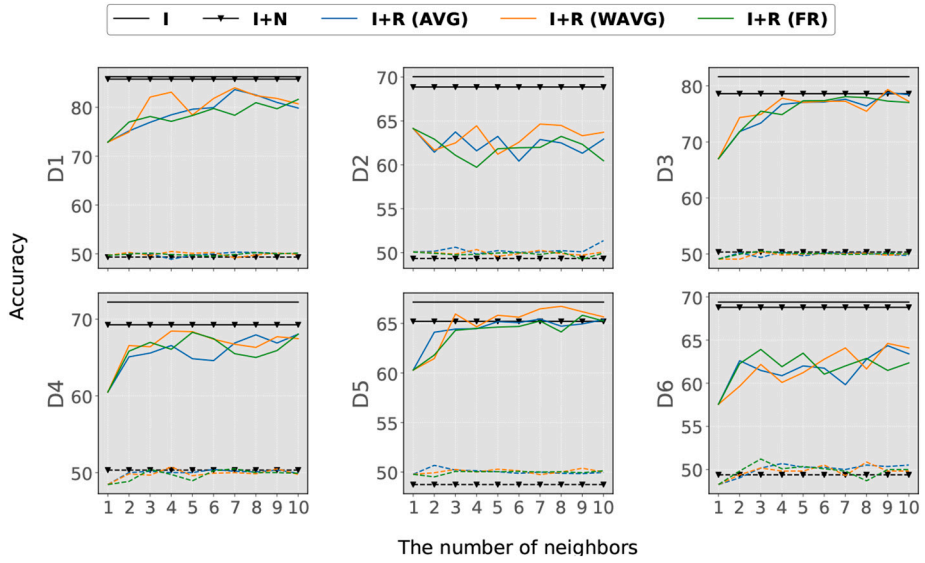

Fig. 3. NAP using non-neighbor context clues. Dotted lines are the context-only counterparts. SFR is excluded from **I** + **R** since random reviews possess no spatial characteristics.

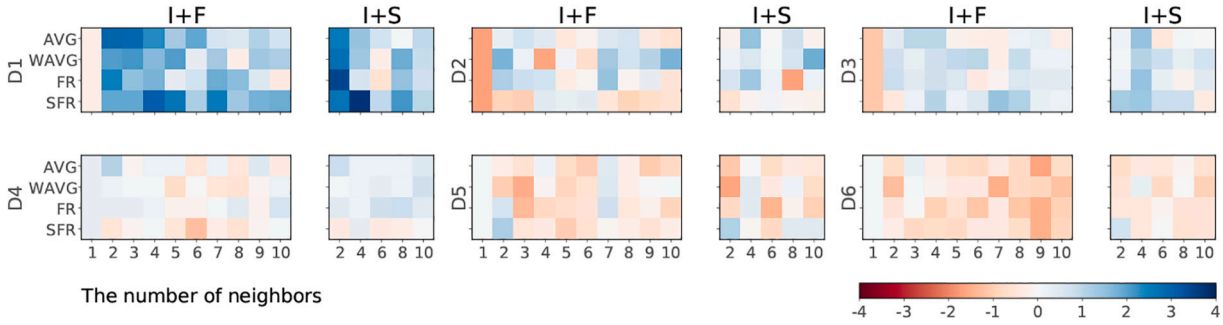

Fig. 4. The increase/decrease in accuracy of **I** + **F** and **I** + **S** compared with **I** + **P**.

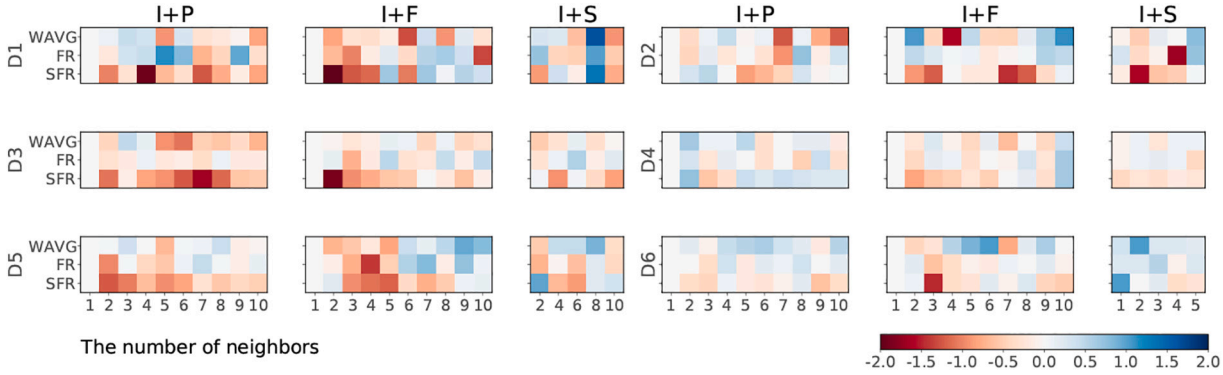

Fig. 5. The increase/decrease in accuracy of WAVG, FR, and SFR compared with AVG.

significant increase in accuracy. We aimed to search for alternative NAP neighbor settings that reduce model complexity while maintaining performance within an acceptable range.

Table 7 listed the alternative neighbor settings. Let p be the neighbor setting in a domain that leads to the highest accuracy q, $\widehat{p}$ is an alternative to p if (1) $\widehat{p}$ uses smaller K values, (2) $\widehat{p}$ uses simpler aggregation schemes, and (3) $|\widehat{q} - q| \leq \delta$. Here, $\delta \in [0, 1]$ ensures that accuracy loss is within 1%. As shown, faster neighbor-aware prediction can be approached using AVG on at most five neighbors at the price of maximum accuracy loss of 0.72%.

5.4. Qualitative analysis on the learned review representations

How a helpfulness prediction model represents reviews reflects its performance. In terms of NAP, review representations refer to the output of the penultimate layer (Eq. (12)) in the network. The goal of helpfulness prediction is to learn a hyperplane that separates the representations for helpful reviews from those for unhelpful ones. Thus, an effective prediction model should represent helpful and unhelpful reviews in the vector space such that one type of the representations can be easily distinguished from another.

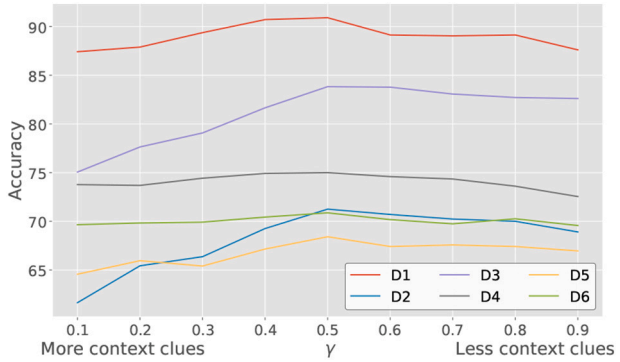

Fig. 6. NAP using different values of the aggregation factor γ.

Table 6
NAP complexity (per epoch). Bias terms are omitted for simplicity.

Scheme	#Floating Point Operations	#Parameters
AVG	mK	0
WAVG	$2mK + 3K + 1$	m
FR	$5mK + 2m$	mK
SFR	$(mK^2 + 11mK)/2 + 2m$	mK

Table 7
Alternative neighbor settings and their decreases in accuracy.

	Neighbor Selection	Neighbor Aggregation	K	δ
D1	$\mathbf{I} + \mathbf{F}$	AVG	6	0.2392
	$\mathbf{I} + \mathbf{F}$	AVG	4	0.7177
D2	$\mathbf{I} + \mathbf{F}$	AVG	7	0.0781
	$\mathbf{I} + \mathbf{S}$	AVG	4	0.2344
D3	$\mathbf{I} + \mathbf{F}$	FR	8	0.0272
	$\mathbf{I} + \mathbf{S}$	FR	4	0.3261
	$\mathbf{I} + \mathbf{S}$	AVG	4	0.4348
D4	$\mathbf{I} + \mathbf{S}$	AVG	6	0.1639
	$\mathbf{I} + \mathbf{F}$	AVG	2	0.4918
D5	$\mathbf{I} + \mathbf{P}$	AVG	5	0.5502
D6	$\mathbf{I} + \mathbf{P}$	WAVG	6	0.0870
	$\mathbf{I} + \mathbf{P}$	WAVG	5	0.1739
	$\mathbf{I} + \mathbf{S}$	WAVG	4	0.5217
	$\mathbf{I} + \mathbf{F}$	AVG	3	0.6957

As an example, we evaluated NAP's performance using test samples from D1 and the first alternative neighbor setting (averaging context clues learned from six following neighbors) in Table 7. We analyzed review representations based on four settings. The first setting is NAP initialized with random weights before model training and the second the GloVe vectors. The third and fourth settings are learned representations for independent and neighbor-aware helpfulness prediction, respectively. We used *t*-SNE [47] to project the 300-dimensional review representations into a two-dimensional plane.

As shown in Fig. 7, helpful and unhelpful reviews in (a) are largely mixed when NAP is initialized with random values. In (b), initializing the embedding table **E** with the GloVe vectors improves the separability between the two types of reviews. Although these two settings show no significant difference in the final accuracy, in our experiments, pre-trained embeddings helped accelerate the training process. Comparing (c) with (d), review representations learned by NAP perform better than those by independent helpfulness prediction in distinguishing helpful from unhelpful reviews. Therefore, using review neighbors can strengthen the predictive power of review helpfulness.

6. Discussions

In Section 6.1, we first summarize the main findings of our work. We also provide theoretical implications for academia in Section 6.2 and managerial implications for the business world in Section 6.3. Finally, Section 6.4 describes current limitations of our work.

6.1. Summary of research findings

We have summarized five key findings from our experimental results. (1) It is feasible to construct effective review representations and context clues from review texts for neighbor-aware helpfulness prediction. (2) The formation of a review's context clues on average engages eight neighbors and tends to treat the neighbors with uneven importance. (3) The major accuracy gains of neighbor-aware prediction occur in close neighbors of a review, with closer neighbors having more influence over its perceived helpfulness. (4) Equally considering up to five closest neighbors of a review for its context clues usually produces weaker but comparable prediction results. (5) The review-neighbor interaction that leads to a review's final helpfulness neither excessively depends on the review nor its neighbors. These findings have filled the gap of existing helpfulness prediction literature and offered insights into developing decision support systems that utilize online user-generated reviews.

6.2. Theoretical implications

This research provides theoretical implications. First, we highlighted and explained the importance of taking a review's neighbors into account when its helpfulness is being perceived, and proposed a new task called neighbor-aware helpfulness prediction. Prior literature mostly assumed a review's helpfulness only depends on information contained in itself [3]. This can easily lead to unreliable prediction [11,13] since human perception is susceptible to sequential bias [5,10]. Recent studies [20–23] attempted to predict helpfulness beyond individual reviews, but assumed customers are conscious of all reviews in a sequence and ignored the review ranking dynamics built in many e-commerce platforms. Our work made a more reasonable proposition that a review's helpfulness results from the interaction between itself and its sequential neighbors. To handle review ranking dynamics, we introduced a dataset wherein reviews are organized in static order overtime. Our work along with the empirical results have demonstrated a promising research

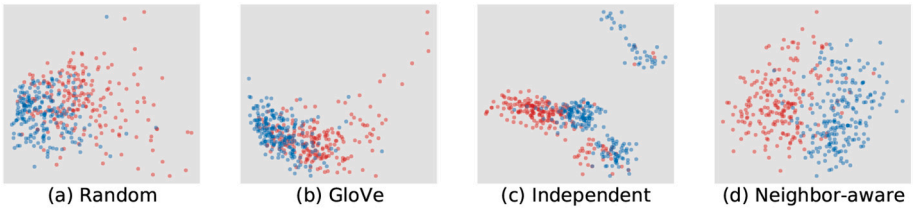

Fig. 7. Document embeddings of helpful (blue) and unhelpful (red) reviews. (a) Random model weights. (b) The embedding table **E** in (a) is initialized by GloVe vectors. (c) Learned weights for independent prediction. (d) Learned weights for neighbor-aware prediction. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

direction in online review helpfulness prediction.

Second, we developed the first end-to-end model for neighbor-aware helpfulness prediction. In contrast to previous work using either individual or all reviews' information, we locally interacted a review with its neighbors for helpfulness modeling. Given a sequence of reviews, a review's neighbors are defined as its adjacent counterparts in the sequence; the review-neighbors interaction combines representations learned from a review and context clues learned from its neighbors. We encoded review representations using deep learning techniques that learn latent semantics directly from review texts. We formulated a two-step learning process for context clue construction: neighbor selection and neighbor aggregation, and respectively designed three and four schemes for each step. The proposed concepts and approaches have laid the methodological foundation for neighbor-aware helpfulness prediction.

Third, we confirmed and revealed how a review is affected by its neighbors during helpfulness perception. To ensure model robustness and result generalizability, we evaluated NAP on real-world reviews collected from six domains and compared NAP with a set of state-of-the-art baselines. Both quantitative and qualitative results showed that considering a review's neighbors can learn more distinguishable and explainable review representations and achieve higher accuracy in helpfulness prediction. We further reported the behavior of review-neighbor interactions by performing ablation studies and sensitivity analysis on NAP's neighbor settings, such as the number of neighbors, neighbor selection strategies, and neighbor aggregation strategies. The summarized research findings of our work can serve as a guidance for future studies to develop prediction models, choose model hyper-parameters, and validate model effectiveness.

6.3. Managerial implications

This research also offers managerial implications. The empirical results of our work showed that customers do not always process reviews in order nor individually even though the reviews are displayed in sequence and written by different users. Instead, the perceived helpfulness of a review usually results from the interaction between its information and surrounding environment – where and how a review is presented – such as review neighbors. This finding suggests several directions in developing effective decision support systems. In a short-term view, display mechanisms can be implemented to briefly inform customers about a review's close neighbors (e.g., text snippets, star ratings) so that customers can interpret the review's rated helpfulness and make purchase decisions more accurately. Meanwhile, tracking systems can be deployed to record customers' review reading and helpfulness rating behaviors. Such data can help existing platforms analyze review-neighbors interactions and how context clues contribute to final review helpfulness in a detailed manner. In a long-term view, new norms of review organization can be designed to alleviate bias in helpfulness perception caused by traditional sequential presentation.

In addition, we observed the assimilation-contrast effects [48] occurred in neighbor-aware helpfulness prediction: a review becomes more helpful if its context clues are learned from neighbors with similar opinions or loses helpfulness with opposite ones. With this finding, decision support systems can help business practitioners maintain their reputation and promote products/services via response management [49]. For instance, existing platforms can remind retailers to address continuous negative reviews to alleviate the assimilation (contrasting) effect on a nearby negative (positive) review. Similarly, managers can appreciate continuous positive reviews to boost their assimilation (contrasting) effect on a nearby positive (negative) review. For platforms that rank reviews, these measures should be acted even more promptly and strategically.

Lastly, as a by-product, NAP can estimate the independent helpfulness of sequentially biased reviews. By setting $\gamma = 1$ in Eq. (12), the trained NAP removes the influence of review neighbors (i.e., context clues) from the predicted helpfulness, which can be thought of as a debiasing process. When handling reviews with ranking dynamics, NAP can be used to adjust reviews' helpfulness based on their neighbors before sorting. Once reviews are sorted, the possibly new neighbors for each review will then be used for adjustment in the next round. NAP can also be extended to other applications that involve human judgments on sequential data, for example, correcting human marking scores [50] on a list of essays.

6.4. Limitations

Our work has two main limitations. The first one concerns the data used for helpfulness modeling. Ideally, NAP would iteratively select a review's neighbors at the time of modeling, which requires multi-snapshot datasets [19] that record reviews' ranking dynamics. As a preliminary study, our work investigated how a review's helpfulness is influenced by its sequential neighbors in one snapshot. We conducted experiments on reviews that are displayed from latest to earliest on the platforms, ensuring that each review's neighbors are relatively static over time. Meanwhile, we are collecting reviews in multiple snapshots for fine-grained neighbor-aware helpfulness prediction.

The other limitation lies in the context clue construction process. The neighbor selection schemes designed in our work have yet to cover many other review reading patterns. For example, customers can read asymmetrical surrounding neighbors, skip certain neighbors of a review while reading, or even read reviews incompletely. Besides, the neighbor aggregation scheme SFR assumed that customers read reviews as per their displayed positions, whereas reading orders can largely vary between customers. These challenges require further investigation into human reading behavior [51] and more flexible methods for context clue construction.

7. Conclusions and future work

We have proposed NAP, the first end-to-end solution for neighbor-aware helpfulness prediction. Given a sequence of online reviews, NAP computes a review's helpfulness based on its content and the context clues learned from its adjacent counterparts in the sequence. Four aggregation schemes have been designed to interact a review with its preceding, following, and surrounding neighbors. Experimental results have confirmed the influence of sequential neighbors on reviews and the effectiveness of NAP. We have further revealed how a review was influenced by its neighbors and summarized insights of our work based on both qualitative and quantitative findings.

We aim to improve NAP in the following directions. In the encoding phase, we plan to design task-specific models to learn deeper semantics from review texts, and include more review-, reviewer-, and product-related metadata into helpfulness modeling. We will also explore novel and more flexible schemes to learn context clues and to model review-neighbors interactions, for instance, a learned rather than specified aggregation factor. Finally, we plan to probe into NAP's performance difference between domains and test how a larger range of neighbors will affect NAP's performance.

CRediT authorship contribution statement

Jiahua Du: Conceptualization, Data curation, Formal analysis, Methodology, Software, Validation, Visualization, Writing - original draft, Writing - review & editing. **Jia Rong**: Methodology, Validation, Supervision. **Hua Wang**: Supervision. **Yanchun Zhang**: Supervision.

Acknowledgements

We would like to thank deeply the editor and reviewers for their valuable time and effort spent on our manuscript and for their constructive comments that help greatly improve the manuscript.

References

[1] R. Murphy, Local Consumer Review Survey. https://www.brightlocal.com/learn/local-consumer-review-survey-2017/, 2017 (accessed 1 April 2021).
[2] H.S. Choi, S. Leon, An empirical investigation of online review helpfulness: a big data perspective, Decis. Support. Syst. 139 (2020) 113403 (12 pages).
[3] G. Ocampo Diaz, V. Ng, Modeling and prediction of online product review helpfulness: a survey, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 698–708.
[4] Z. Zhang, Q. Ye, R. Law, Y. Li, The impact of e-word-of-mouth on the online popularity of restaurants: a comparison of consumer reviews and editor reviews, Int. J. Hosp. Manag. 29 (2010) 694–700.
[5] S. Sridhar, R. Srinivasan, Social influence effects in online product ratings, J. Mark. 76 (2012) 70–88.
[6] G. Askalidis, S.J. Kim, E.C. Malthouse, Understanding and overcoming biases in online review systems, Decis. Support. Syst. 97 (2017) 23–30.
[7] A.R. Camilleri, The importance of online reviews depends on when they are presented, Decis. Support. Syst. 133 (2020) 113307 (11 pages).
[8] A. Jha, S. Shah, Social influence on future review sentiments: an appraisal-theoretic view, J. Manag. Inf. Syst. 36 (2019) 610–638.
[9] Y. Zhu, M. Liu, X. Zeng, P. Huang, The effects of prior reviews on perceived review helpfulness: a configuration perspective, J. Bus. Res. 110 (2020) 484–494.
[10] R.T. Sikora, K. Chauhan, Estimating sequential bias in online reviews: a kalman filtering approach, Knowled Based Syst. 27 (2012) 314–321.
[11] W.W. Moe, M. Trusov, The value of social dynamics in online product ratings forums, J. Mark. Res. 48 (2011) 444–456.
[12] L. Qiu, W. Wang, The effects of message order and information chunking on ewom persuasion, in: Proceedings of the 15th Pacific Asia Conference on Information Systems, Brisbane, Australia, 2011, pp. 1–13.
[13] L. Page, K. Page, Last shall be first: a field study of biases in sequential performance evaluation on the idol series, J. Econ. Behav. Organ. 73 (2010) 186–198.
[14] B. Gao, N. Hu, I. Bose, Follow the herd or be myself? An analysis of consistency in behavior of reviewers and helpfulness of their reviews, Decis. Support. Syst. 95 (2017) 1–11.
[15] J.B. Walther, Y.J. Liang, T. Ganster, D.Y. Wohn, J. Emington, Online reviews, helpfulness ratings, and consumer attitudes: An extension of congruity theory to multiple sources in web 2.0, J. Comput.-Mediat. Commun. 18 (2012) 97–112.
[16] Y.-C.C. Ho, J. Wu, Y. Tan, Disconfirmation effect on online rating behavior: a structural model, Inf. Syst. Res. 28 (2017) 626–642.
[17] S. Quaschning, M. Pandelaere, I. Vermeir, When consistency matters: the effect of valence consistency on review helpfulness, J. Comput.-Mediat. Commun. 20 (2015) 136–152.
[18] L. Qiu, J. Pang, K.H. Lim, Effects of conflicting aggregated rating on ewom review credibility and diagnosticity: the moderating role of review valence, Decis. Support. Syst. 54 (2012) 631–643.
[19] R. Sipos, A. Ghosh, T. 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, , ACM, Seoul, Korea, 2014, pp. 337–348.
[20] S. Zhou, B. Guo, The order effect on online review helpfulness, Decis. Support. Syst. 93 (2017) 77–87.
[21] J.E. Fresneda, D. Gefen, A semantic measure of online review helpfulness and the importance of message entropy, Decis. Support. Syst. 125 (2019) 113117 (11 pages).
[22] Y. Hong, J. Lu, J. Yao, Q. Zhu, G. 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, ACM, Portland, Oregon, USA, 2012, pp. 495–504.
[23] Y. Lu, P. Tsaparas, A. Ntoulas, L. Polanyi, Exploiting social context for review quality prediction, in: Proceedings of the 19th International Conference on World Wide Web, ACM, Raleigh, North Carolina, USA, 2010, pp. 691–700.
[24] E. Eryarsoy, S. Piramuthu, Experimental evaluation of sequential bias in online customer reviews, Inf. Manag. 51 (2014) 964–971.
[25] H. Baek, J. Ahn, Y. Choi, Helpfulness of online consumer reviews: Readers' objectives and review cues, Int. J. Electron. Commer. 17 (2012) 99–126.
[26] Y. Kim, Convolutional neural networks for sentence classification, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Doha, Qatar, 2014, pp. 1746–1751.
[27] D. Clevert, T. Unterthiner, S. Hochreiter, Fast and accurate deep network learning by exponential linear Units (ELUs), in: Proceedings of the 4th International Conference on Learning Representations, San Juan, Puerto Rico, 2016, pp. 1–14.
[28] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, P. Kuksa, Natural language processing (almost) from scratch, J. Mach. Learn. Res. 12 (2011) 2493–2537.
[29] L. Zhang, Q. Yan, L. Zhang, A text analytics framework for understanding the relationships among host self-description, trust perception and purchase behavior on airbnb, Decis. Support. Syst. 133 (2020) 1–10.
[30] S. Mitra, M. Jenamani, Helpfulness of online consumer reviews: a multi-perspective approach, Inf. Process. Manag. 58 (2021) 102538 (17 pages).
[31] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Proceedings of the 26th International Conference on Neural Information Processing Systems, Lake Tahoe, Nevada, USA, 2013, pp. 3111–3119.
[32] Y. Ma, Z. Xiang, Q. Du, W. Fan, Effects of user-provided photos on hotel review helpfulness: an analytical approach with deep leaning, Int. J. Hosp. Manag. 71 (2018) 120–131.
[33] W. Kwon, M. Lee, K.-J. Back, Exploring the underlying factors of customer value in restaurants: a machine learning approach, Int. J. Hosp. Manag. 91 (2020) 102643 (13 pages).
[34] L. Men, N. Ilk, X. Tang, Y. Liu, Multi-disease prediction using lstm recurrent neural networks, Expert Syst. Appl. 177 (2021) 114905 (11 pages).
[35] C.C. Chen, Y.-D. Tseng, Quality evaluation of product reviews using an information quality framework, Decis. Support. Syst. 50 (2011) 755–768.
[36] J. Du, J. Rong, S. Michalska, H. Wang, Y. Zhang, Feature selection for helpfulness prediction of online product reviews: an empirical study, PLoS One 14 (2019) 1–26.
[37] S. Krishnamoorthy, Linguistic features for review helpfulness prediction, Expert Syst. Appl. 42 (2015) 3751–3759.
[38] M. Malik, A. Hussain, An analysis of review content and reviewer variables that contribute to review helpfulness, Inf. Process. Manag. 54 (2018) 88–104.
[39] P.D. Turney, P. Pantel, From frequency to meaning: vector space models of semantics, J. Artif. Intell. Res. 37 (2010) 141–188.
[40] Y.-C. Chang, C.-H. Ku, C.-H. Chen, Using deep learning and visual analytics to explore hotel reviews and responses, Tour. Manag. 80 (2020) 104129 (22 pages).
[41] J. Pennington, R. Socher, C. Manning, GloVe: Global Vectors for Word Representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Doha, Qatar, 2014, pp. 1532–1543.
[42] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Proceedings of the 13th International Conference on Artificial Intelligence and Statistics, Sardinia, Italy, 2010, pp. 249–256.
[43] PowerReviews, Proven Power of Ratings & Reviews: A Report. https://www.powerreviews.com/insights/proven-power-of-ratings-and-reviews/, 2014 (accessed 1 April 2021).
[44] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, in: Proceedings of the 3rd International Conference on Learning Representations, San Diego, California, USA, 2015, pp. 1–15.
[45] R. Dror, G. Baumer, S. Shlomov, R. Reichart, The hitchhiker's guide to testing statistical significance in natural language processing, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 1383–1392.
[46] G. Askalidis, E.C. Malthouse, The value of online customer reviews, in: Proceedings of the 10th ACM Conference on Recommender Systems, ACM, Boston, Massachusetts, USA, 2016, pp. 155–158.
[47] L.V.D. Maaten, G. Hinton, Visualizing data using t-sne, J. Mach. Learn. Res. 9 (2008) 2579–2605.
[48] K. Wilcox, A.L. Roggeveen, D. Grewal, Shall i tell you now or later? Assimilation and contrast in the evaluation of experiential products, J. Consum. Res. 38 (2011) 763–773.
[49] K.L. Xie, Z. Zhang, Z. Zhang, The business value of online consumer reviews and management response to hotel performance, Int. J. Hosp. Manag. 43 (2014) 1–12.
[50] H. Zhao, B. Andersson, B. Guo, T. Xin, Sequential effects in essay ratings: evidence of assimilation effects using cross-classified models, Front. Psychol. 8 (2017) 933 (10 pages).
[51] E. Maslowska, C.M. Segijn, K.A. Vakeel, V. Viswanathan, How consumers attend to online reviews: an eye-tracking and network analysis approach, Int. J. Advert. 39 (2020) 282–306.

Jiahua Du obtained his PhD in Computer Science from the Institute for Sustainable Industries & Liveable Cities at Victoria University in 2020. He obtained his BSc and MSc in Computer Science from South China Normal University in 2012 and 2015. His research interests focus on machine learning, deep learning, data mining, natural language processing, social media analysis, and their applications. He is particularly interested in quality evaluation, knowledge discovery, and content recommendation on online user-generated reviews.

Jia Rong joined Monash University as a lecturer in Data Science with the Faculty of Information Technology from March 2019. She obtained her Bachelor, Honours and PhD in Computer Science and Information Technology from Deakin University. Since 2007, she has taught more than twenty subjects at Victoria University, Deakin University, MIBT/Deakin College, Australian Technical and Management College and Central Queensland University. Her research lies at the joint area of pattern recognition, text analysis, big data analysis and social network analysis. She is particularly interested in supervised deep learning models for online review analysis, product and service recommendations. She also works in the area of big data validation and processing.

Hua Wang is a Professor at Victoria University. He was a professor at the University of Southern Queensland before he joined Victoria University in 2014. He has expertise in Big Data, Health Informatics, Cloud Computing, Cyber Security, Artificial Intelligence. He has received four Australian Research Council (ARC) Discovery grants and two Linkage grants since 2006. He has also received two Japan–Australian grants, one German–Australian grant, one Norway Government grant and grants from Hong Kong Chinese University and Hong Kong City University. He has published 258 refereed scholar papers in data security, access control, data mining, database, privacy preserving and Web services. Representative publications are on TOIT, TOIS, TKDD, TDSC, TKDE, T-ASE and World Wide Web; and Proceedings of ACL, AAAI, CIKM, ICDE, ICDM, and PAKDD.

Yanchun Zhang is a Professor and Director of Data Science and AI program with the Institute for Sustainable Industries & Liveable Cities, Victoria University. He obtained the PhD degree in Computer Science from The University of Queensland in 1991. His research interests include databases, data mining, web services and e-health. He has published over 300 research papers in international journals and conference proceedings including TOCHI, TKDE, VLDBJ, SIGMOD, and ICDE, and books and journal special issues in the related areas. He is a founding editor and editor-in-chief of World Wide Web Journal and Health Information Science and Systems Journal.

Yanchun Zhang is a Professor and Director of Data Science and AI program with the Institute for Sustainable Industries & Liveable Cities, Victoria University. He obtained the PhD degree in Computer Science from The University of Queensland in 1991. His research interests include databases, data mining, web services and e-health. He has published over 300 research papers in international journals and conference proceedings including TOCHI, TKDE, VLDBJ, SIGMOD, and ICDE, and books and journal special issues in the related areas. He is a founding editor and editor-in-chief of World Wide Web Journal and Health Information Science and Systems Journal.