Data Science and Engineering (2020) 5:261–279
https://doi.org/10.1007/s41019-020-00133-1

An Interactive Network for End-to-End Review Helpfulness Modeling

Jiahua Du[1] · Liping Zheng[2] · Jiantao He[3] · Jia Rong[4] · Hua Wang[1] · Yanchun Zhang[1]

Received: 9 March 2020 / Revised: 23 May 2020 / Accepted: 11 June 2020 / Published online: 26 June 2020

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

Keywords Review helpfulness · Review texts · Star ratings · Text–rating interaction

1 Introduction

Online reviews play an important role in the e-commerce ecosystem. Currently, online buyers highly rely on collective wisdom to make informed purchase decisions. A recent survey [43] shows that over 8 of 10 customers read reviews for online retailers. The reviews also help manufactures collect user feedback and improve products. Nevertheless, the quality of user-generated reviews is uneven [34], susceptible to customers' background, tolerance of product deficiencies, moods at the time of writing, to name a few. As the number of reviews grows, locating useful information becomes increasingly challenging. Many e-commerce platforms gather user voting on review helpfulness for quality assessment. Still, the voting data are scarce in practice and even missing in less popular products.

Helpfulness prediction aims to identify and recommend high-quality reviews to customers in an automatic manner. The previous literature [2, 22, 44] largely employs review texts and star ratings for the task. The rationale lies in their ubiquitousness in contemporary online shopping platforms and their importance to review helpfulness modeling. Review texts qualitatively describe reviewers' opinions toward product properties. The textual content contains rich information [16], which is an ideal source [11] for learning helpfulness information. On the other hand, star ratings [40] provide a more straightforward form to quantify reviewers' opinions. The valence (positive or negative) [66] and extremity [15, 45, 57] of ratings are shown to have considerable impact on review helpfulness.

More importantly, the (in)consistency [52, 60, 62] between review texts and star ratings can also affect a consumer's helpfulness perception. The text of a review and its

✉ Jiahua Du
yukachan.d@gmail.com

Liping Zheng
17210240265@fudan.edu.cn

Jiantao He
hejiantao@gzmtr.com

Jia Rong
jiarong@acm.org

Hua Wang
hua.wang@vu.edu.au

Yanchun Zhang
yanchun.zhang@vu.edu.au

1 Institute of Sustainable Industries and Liveable Cities, Victoria University, Melbourne, VIC, Australia

2 School of Computer Science and Technology, Fudan University, Shanghai, China

3 Guangzhou Metro Group Co., Ltd., Guangzhou, Guangdong, China

4 Faculty of Information Technology, Monash University, Clayton, VIC, Australia

accompanying star rating can be thought of as the qualitative and quantitative aspects [68] of the same user experience. Normally, customers expect the overall opinion of review content to be aligned with the rating [23] during perusal. In practice, however, a review's rating does not necessarily reflect what is mentioned in the content [65] due to the subjectivity of ratings [30, 56]. As a toy example, Fig. 1 shows two reviews with the same comments, but different ratings. In review (b), the mismatching opinions may be considered careless, over-subjective, or being ironic. Such inconsistency is likely to cause confusion and diminish the trustworthiness and thus helpfulness of a review.

Existing methods combine review texts and star ratings for helpfulness modeling to imitate customers measuring the (in)consistency. However, star ratings are used in a way that has either little representation capacity or limited interaction with review texts. In most studies [17, 54, 58], review texts are represented in a high-dimensional feature space, whereas star ratings are used directly. The scalar representation limits the capacity of rating information as well as its influence on review texts. To enlarge encoding capacity, [51] treats star ratings as the final word of its text. The combination is done by learning star embeddings as part of review text encoding, using convolution neural networks (CNNs) as encoders. CNNs operationalize sliding windows on a text to learn features from consecutive words. Under this setting, a star rating only locally interacts with the last few words of a text and thus has limited interaction. Also, rating information may lose during text encoding due to the max pooling nature in CNNs. As a result, the existing methods are far from fully utilizing rating information.

This work further utilizes rating information for helpfulness modeling. An end-to-end architecture is proposed to learn text–rating interaction (TRI). To enable equivalent representation capacity, TRI maps review texts and star ratings into feature vectors of the same dimensionality. To enlarge the text–rating interaction during combination, text and rating embeddings are first separately learned and then combined. Different from [51] that learns rating vectors as part of content encoding, the encoding of star ratings is decoupled from that of review texts. As a result, rating information can interact with all words in a review text. The decoupling also helps the rating information remain intact and maintain its global influence on review content. The (in)consistency between review texts and star ratings is then captured via the element-wise interaction between content and rating vectors. During the interaction, TRI further adopts gating mechanisms to adaptively learn the amount of rating information needed by review content.

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

The remainder of the paper is organized as follows. Section 2 surveys the related work. Section 3 gives the problem statement of interactive helpfulness modeling. Section 4 presents TRI and its learning components. Section 5 describes experiment settings used to evaluate TRI against a series of state-of-the-art methods. Section 6 demonstrates the effectiveness of TRI, conducts detailed ablation studies of the TRI components, and discusses the behavior of TRI via a series of qualitative analysis tasks. Finally, Sect. 7 concludes the paper.

2 Related Work

Review texts have been used as the main source for helpfulness prediction due to the rich information. Combining review texts and star ratings for helpfulness modeling is also gaining increasing attention. This section introduces existing methods using sole review content (Sect. 2.1) and the conjunction of review content and star ratings (Sect. 2.2), respectively.

2.1 Content-Based Helpfulness Prediction

Various models [17, 25, 35, 38, 64] have been proposed to identify helpful reviews. The mainstream solution [44] is to design feature patterns from review texts, review metadata, and social networks of reviewers. Such methods, albeit effective, are often product- and domain-dependent. The feature preparation process also requires prior knowledge and

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

(a) Positive comment with positive rating

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

(b) Positive comment with negative rating

Fig. 1 Consistency between review texts and star ratings can affect helpfulness perception

tremendous amounts of time. Moreover, the extracted features often suffer from a varying degree of multi-collinearity [46] (i.e., feature redundancy), which introduces unexpected noise into the feature space.

The emergence of deep learning [32, 36, 37, 48] is bringing new paradigms into helpfulness prediction. In particular, CNNs [26, 27] have shown the feasibility in modeling helpfulness information. Saumya et al. [55] employ a two-layer CNN framework. In [4], helpfulness prediction is considered as a cross-domain task. The authors learn document embeddings for individual reviews, upon which three separate CNN layers are built to perform knowledge transferring. Chen et al. [3] extend [4] to study multi-domain helpfulness prediction. The authors assume that words in a review contribute diversely to helpfulness and propose the embedding-gated CNN (EG-CNN) to identify important/unimportant words in a review. The aspect distribution [63] of a review is also incorporated into representation learning.

Following the previous work, TRI develops a CNN-based architecture to learn features from review texts. Similar to [3], gating mechanisms are used during content representation learning. Differently, the gates in TRI are not only used to identify word importance but also to combine word embeddings and the convoluted features for multi-granularity content representations.

2.2 Interaction Between Review Texts and Star Ratings

The past decade has seen a large body of studies [2, 22, 44] relying on both review texts and star ratings for helpfulness prediction. In most of the feature engineering approaches [5, 6, 19, 23], rating information is used in conjunction with review texts by concatenating learned content representations and raw rating values. Several studies [37, 38, 65] consider star ratings as a moderating factor to interact with review texts. In addition to linear rating information, the quadratic term [1, 29, 42] of star ratings is used to validate the influence of rating extremity on the perceived helpfulness.

More recently, deep learning techniques have been used to model more sophisticated text–rating interactions. Qu et al. [51] proposed two CNN variants to combine review texts and star ratings. In the first combination method (CM1), a review's star rating and its learned content representation are concatenated following conventional feature engineering. The second combination method (CM2) treats a star rating as the last word of a review. As such, star ratings are converted into vectors of the same dimensionality of word embeddings. The word embeddings and star embeddings are then concatenated for learning content representations.

Fan et al. [13] formulate review helpfulness prediction as a multitask neural learning (MTNL) problem. Specifically, a character-level CNN framework is employed to learn continuous features from review texts. The content representation of a review results from the weighted average of the feature maps using attention mechanisms. The learned representation is then used to perform two prediction tasks simultaneously: the classification of review helpfulness and the regression of the accompanying star rating. The same training objectives are adopted by [14].

Figure 2 depicts the three main methods utilizing review texts and star ratings. In (a), star ratings and learned content representations are concatenated, which cannot capture the mutual interaction between the two features. Even using ratings as a moderator, the weak interaction is constrained by the scalar representation of star ratings. In (b), star ratings are converted into rating embeddings to enlarger encoding space. Still, rating information has limited interaction with review texts and may lose [24] during the content encoding phase. For example, CM2 interacts ratings with texts through convolution and max pooling. In two extreme cases [5], rating information can dominate the whole representation or do not influence at all. Conversely, star ratings in (c) are used as one of the outputs to be predicted. Such methodology is arguably counterintuitive because it assumes customers are unaware of rating information when deciding review helpfulness.

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

Inspired by [12, 51, 68], TRI embeds star ratings to enlarge the encoding space for rating information. Different from [51], TRI decouples the representation learning of review texts from that of star ratings to avoid possible loss of rating information. To the best of our knowledge, TRI is the first work that takes into account both the encoding and interactive capability of rating information for helpfulness modeling.

3 Problem Definition

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

Let D be a collection of raw online reviews. Each review $d = (r, s, y) \in D$ is a tuple of its text content s, the accompanying star rating r, and the helpfulness label $y \in \{0, 1\}$. The

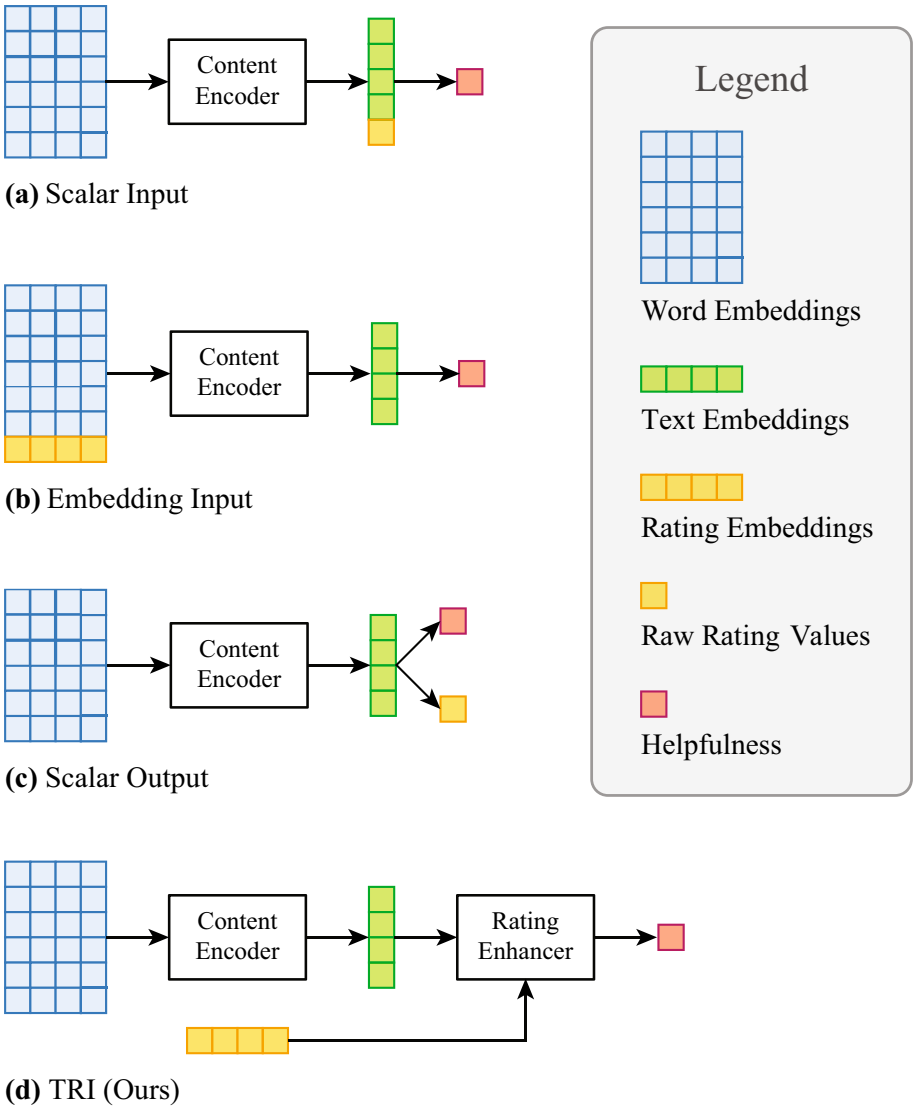

Fig. 2 Existing approaches combining review texts and star ratings

label $y = 0$ indicates an unhelpful review and $y = 1$ helpful. The goal of helpfulness prediction is to learn a classification model F parameterized by θ:

$$F(s, r; \theta) \longrightarrow \hat{y}. \tag{1}$$

The model takes as inputs review texts and star ratings, and learns helpfulness information from their interaction. For each review, the model then produces a helpfulness label $\hat{y}$ that approximates the actual helpfulness y of the review.

4 Text–Rating Interaction Networks

Figure 3 illustrates the TRI architecture. Given a review $d = (r, s, y)$, TRI starts with transforming the text s into an embedding matrix $\mathbf{X}$ and star rating r into an embedding $\mathbf{e}'_r$. Two TRI components are introduced: The content encoder learns content representations $\mathbf{h}$ from $\mathbf{X}$, whereas the rating enhancer learns adaptive rating representations $\mathbf{r}'$ from $\mathbf{e}'_r$. The two representations are then interacted to jointly learn the rating-enhanced document embedding $\mathbf{h}'$ for helpfulness prediction. The following subsections will give more details of the two learning components.

4.1 Content Encoder

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

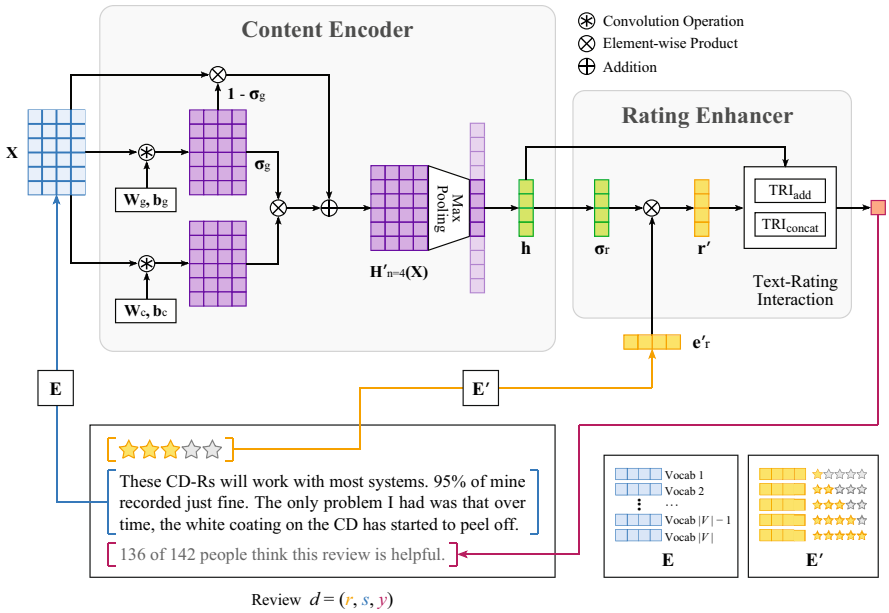

Fig. 3 TRI architecture

embedding matrix $\mathbf{X} \in \mathbb{R}^{N \times d}$ by simply stacking the embedding of the constituent words:

$$\mathbf{X} = [\mathbf{e}_{x_1}, \mathbf{e}_{x_2}, \ldots, \mathbf{e}_{x_N}]. \tag{2}$$

The embedding matrix $\mathbf{X}$ of a text is used for hidden semantic extraction. Previous work has predominantly used recurrent neural networks (RNNs) [33] for text encoding due to the sequential nature. In RNNs, a memory of occurring information in a sequence is maintained. Training such memory is computationally inefficient as it cannot be parallelized over sequential tokens. One alternative is using gated linear units (GLUs) [10], which allows for parallelization while maintaining a large range of memory. As such, GLUs can be thought of as a faster implementation that approximates the behavior of RNNs.

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

$$\mathbf{H}(\mathbf{X}) = (\mathbf{X} * \mathbf{W}_c + \mathbf{b}_c) \otimes \boldsymbol{\sigma}_g, \tag{3}$$

$$\boldsymbol{\sigma}_g = \sigma(\mathbf{X} * \mathbf{W}_g + \mathbf{b}_g), \tag{4}$$

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

The use of GLUs for encoding review texts is advantageous. First, GLUs facilitate the training process by allowing gradients to flow through the encoder layers. During back propagation, the first addend of the gradient $\nabla \mathbf{H}(\mathbf{X}) = \nabla(\mathbf{X} * \mathbf{W}_c + \mathbf{b}_c) \otimes \boldsymbol{\sigma}_g + (\mathbf{X} * \mathbf{W}_c + \mathbf{b}_c) \otimes \boldsymbol{\sigma}_g' \nabla (\mathbf{X} * \mathbf{W}_c + \mathbf{b}_c)$ provides a linear path that maintains the scale of the activated gating units. Such a structure reduces the gradient vanishing problem in neural networks as more layers are stacked. The linear path can also be thought of as a multiplicative skip connection [20] between encoder layers. Secondly, the values of $\boldsymbol{\sigma}_g \in [0, 1]$ enable gating mechanisms on the convoluted features. In this case, each of the encoded (convoluted) word embeddings is bound with a gate indicating different word importance. This resembles the use of gated word embeddings in [3] for multi-granularity text features. Thirdly, $\boldsymbol{\sigma}_g$ are further utilized to merge the word embeddings (low-level information) and feature maps (high-level information):

$$\mathbf{H}'(\mathbf{X}) = \mathbf{H}(\mathbf{X}) + (1 - \boldsymbol{\sigma}_g) \otimes \mathbf{X}. \tag{5}$$

Here, the gates $\boldsymbol{\sigma}_g$ estimate the ratio of low- and high-level information required. From the perspective of GRU [7], the combination can also be thought of as determining how much new information $\mathbf{H}(\mathbf{X})$ is used to update the previous memory $\mathbf{X}$ at each time step. Setting the values of $\boldsymbol{\sigma}_g$ to 1 considers only the feature maps. In contrast, $\boldsymbol{\sigma}_g = 0$ indicates the exclusive use of the word embeddings.

In TRI, kernels of patch size $n = \{3, 4, 5\}$ are used simultaneously to learn hidden semantics from n-grams in a review. Column-wise max-overtime pooling [8] is then

applied to obtain the most salient features. Finally, the pooled features are concatenated and projected via learnable parameters $\mathbf{W}_h \in \mathbb{R}^{3d \times m}$:

$$\mathbf{h} = \left[\max\{\mathbf{H}'_{n=3}(\mathbf{X})\}, \max\{\mathbf{H}'_{n=4}(\mathbf{X})\}, \max\{\mathbf{H}'_{n=5}(\mathbf{X})\}\right]\mathbf{W}_h, \tag{6}$$

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

4.2 Rating Enhancer

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

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

The rating embedding $\mathbf{e}'_r$ is set to have the same dimensionality as the content representation $\mathbf{h}$ to perform element alignment. As discussed, review texts and star ratings can be thought of as two measurements of the same user experience. While both measurements take different forms of output (i.e., words versus a scalar), the latent evaluation criteria that lead to the decisions are highly similar. This work hypothesizes that such criteria are reflected by individual embedding dimensions. Thus, aligning $\mathbf{h}$ and $\mathbf{e}'_r$ forces the network to encode each criterion into the same dimension in both embeddings.

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

$$\boldsymbol{\sigma}_r = \sigma(\mathbf{W}_r^{\top}\mathbf{h} + \mathbf{b}_r), \tag{7}$$

$$\mathbf{r}' = \mathbf{e}'_r \otimes \boldsymbol{\sigma}_r. \tag{8}$$

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

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

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

$$\mathbf{h}' = \mathbf{h} + \mathbf{r}'. \tag{9}$$

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

$$\mathbf{h}' = \left[\mathbf{h}, \mathbf{r}'\right]. \tag{10}$$

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

4.3 Training Objective

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

$$\hat{y} = \sigma(\mathbf{W}_o^{\top}\mathbf{h}' + b_o), \tag{11}$$

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

$$\mathscr{L} = -\frac{1}{M}[\mathbf{y}^\top \log(\hat{\mathbf{y}}) + (1-\mathbf{y})^\top \log(1-\hat{\mathbf{y}})], \quad (12)$$

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

5 Experiment Settings

This section conducts extensive experiments to quantitatively and qualitatively evaluate TRI. Section 5.1 gives a brief introduction to the datasets and preprocessing steps used throughout the experiments. In Sect. 5.2, the baselines using both traditional machine learning algorithms and state-of-the-art deep learning methods are described for performance comparison. Section 5.3 presents hyperparameters for training TRI and the baseline models.

5.1 Datasets

TRI is evaluated on the public Amazon 5-Core dataset [21]. The original dataset consists of 24 domains, covering 142.8 million reviews collected between May 1996 and July 2014. Amazon is the largest Internet retailer that has accumulated large-scale user-generated reviews. The helpfulness of the reviews is rated by online customers, offering an ideal source for the review helpfulness prediction task. Amazon product reviews are predominantly used and analyzed in previous studies. Thus, adopting Amazon reviews allows for fair comparison with previous studies. The analysis results can also provide practical insights into online business and user-generated content quality evaluation.

The six largest domains are selected for evaluation, including Apps for Android, Video Games, Electronics, CDs and Vinyl, Movies and TV, and Books. A large number of online reviews ensure sufficient training data for the TRI architecture. For simplicity, the first domain is called D1, the second D2, and so on. Table 1 shows a random review from Video Games. Each review contains (1) the ID of a reviewed product, (2) the helpfulness information, namely user-provided helpful and unhelpful votes, (3) a star rating, (4) the published date, week, and time, (5) the ID and name of the reviewer, and (6) a text composed of a summary headline and detailed comments on the product. This paper focuses on using the review text and star rating of a review.

The vote distributions presented in Fig. 4 reveal similar patterns that a proportion of reviews have relatively few votes. Figure 5 demonstrates the review length (i.e., the number of words) distributions. Overall, the length of most reviews is within a certain range, with a small number of outliers being unusually long. Further analysis in Table 2 reveals that the longest review in a domain is on average ten times longer than 90% of the rest. Figure 6 presents the review rating distributions. As shown, customers tend to give positive feedback, with five-star (four- and five-star) ratings accounting for over half (70%) of the reviews. This phenomenon is identified as positivity bias [39] in accordance with many existing studies.

The following preprocessing is applied. (1) Blank and non-English reviews are filtered. (2) For identical and nearly identical reviews [9], only the ones with the highest number of votes are retained. The sameness detection follows [25] by examining whether two reviews share more than 80% of their bigram occurrences. (3) To alleviate biases caused by the "words of few mouths" [53, 67] phenomenon, reviews with less than 10 votes are skipped. (4) The remaining reviews are then lowercased and tokenized, followed by removing the articles "a," "an," and "the." Further stop word removal is not considered since some stop words (e.g., negation) can be useful in building review helpfulness. (5) To accelerate the training process, the sequence length N for each domain is set to the one whose word count is larger than

Table 1 Example Amazon review composition

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

Typos and capitalization in the original reviews are intentionally preserved

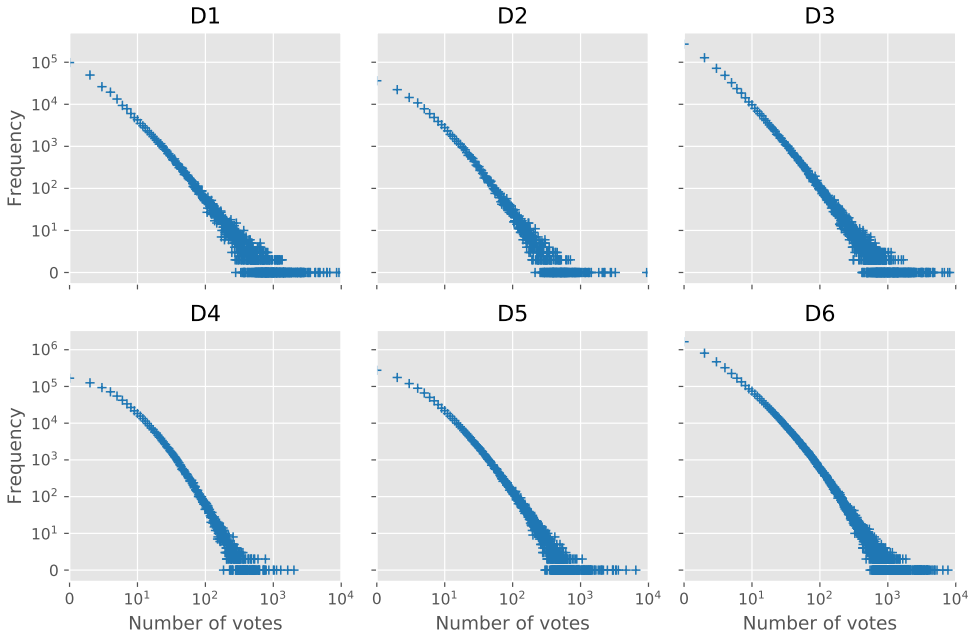

Fig. 4 Review vote distributions

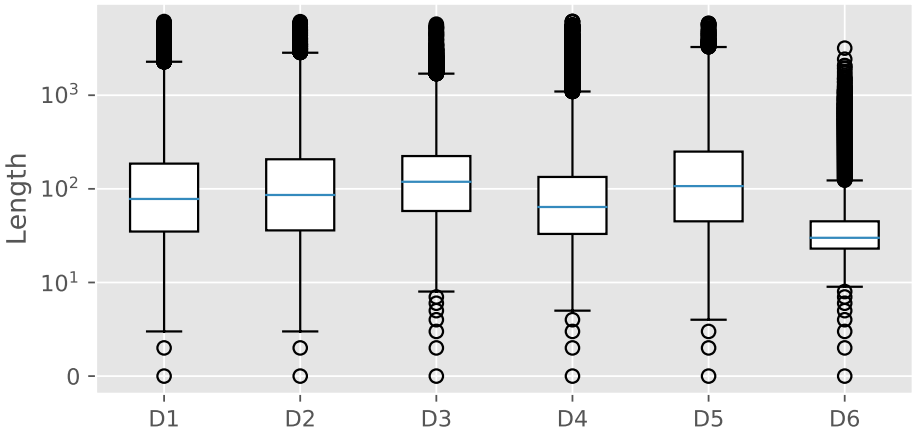

Fig. 5 Review length distributions

Table 2 Review length difference

Length	D1	D2	D3	D4	D5	D6
At 100th percentile (maximum)	1641	5531	5143	4813	5234	5441
At 90th percentile	109	757	546	469	549	517
Multiples	15.06	7.31	9.42	10.26	9.53	10.52

90% of the reviews. In our preliminary experiments, training vanilla CNNs on the original and truncated reviews have little difference in performance, but the latter on average takes one-tenth of the parameters of the former.

The preprocessed reviews are labeled as follows. Amazon allows for dichotomous options for review helpfulness. Following prior research [17, 31, 37], a review is labeled as helpful if at least 60% of its received votes are helpful, and labeled as unhelpful otherwise. The reviews in one class are then randomly sampled to have the same number of reviews as the other for class balance.

For each domain, reviews are partitioned using the stratified random split scheme. The whole collection of reviews is first shuffled. Subsequently, 80, 10, and 10% of the reviews are randomly selected, respectively, to build the training, validation, and test set. During the selection, the percentage of samples for each class is preserved. Throughout the paper, TRI and the baselines are trained

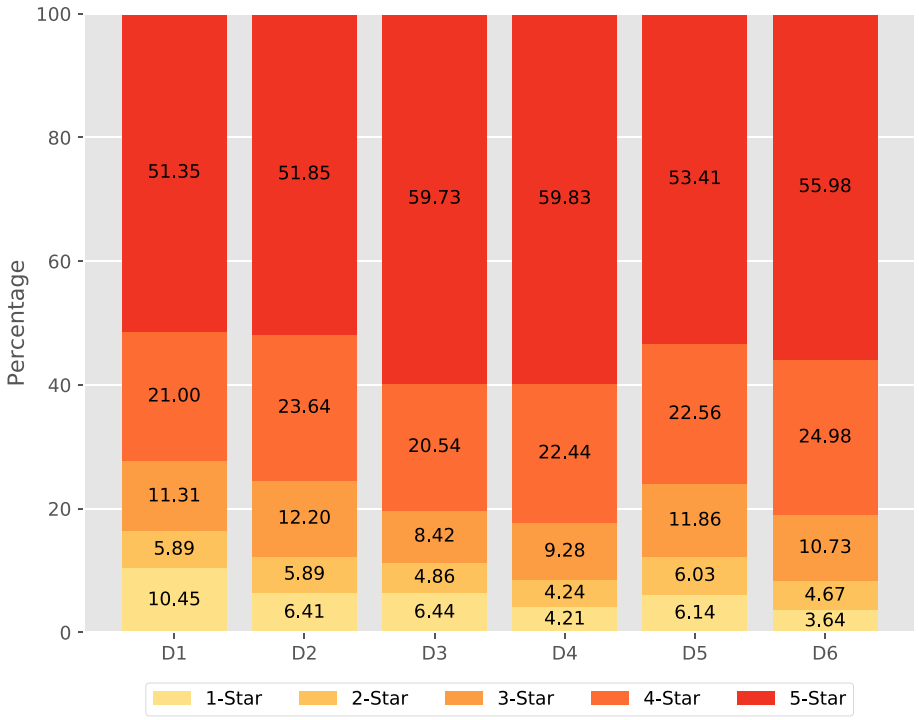

Fig. 6 Review rating distributions

on the training set, tuned on the validation set, and evaluated on the test set treated as unseen data.

To reduce computational consumption, only the top 30 k frequent terms are kept as vocabulary. In the preliminary experiments on small datasets D1 and D2, the performance of using the top 30 k terms is similar to that of using the full vocabulary, showing the feasibility in training models without less frequent terms. Finally, numeric values in the reviews are replaced by < NUM >, whereas out-of-vocabulary words are altered by < UNK >.

Table 3 demonstrates the descriptive statistics of the six domains sorted by data size in ascending order. From left to right, the six types of statistics are (1) the total number of reviews, (2) the total number of tokens, (3) the average number of tokens per review, (4) the total number of sentences, (5) the average number of sentences per review, and (6) the average number of tokens per sentence. A token is clarified as an instance of a sequence of characters, which results from tokenization in the preprocessing steps. The concept of a token is similar to that of a word except that a token does not necessarily have to be a valid English word.

5.2 Baseline Methods

TRI is benchmarked against twelve baselines, including seven traditional machine learning methods and five state-of-the-art deep learning architectures. Note that the PRH-Net model [14] uses extra product information for training, which is unfair to TRI and thus skipped.

- TFIDF + SVM: Unigrams have been proved robust and effective in many text mining applications. This baseline trains linear SVM classifiers on TFIDF representations of review unigrams, where terms with document frequency fewer than 1% of the training samples are ignored.
- Recent word embeddings learned from shallow neural networks also show promising performance. Following [11], three types of pretrained embeddings are used. SVM classifiers are then trained on review representa-

Table 3 Descriptive statistics of the balanced domains after preprocessing

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

tions. The embedding of a review is the average of that of its constituent words, where out-of-vocabulary words are ignored.

  - SGNS + SVM: This baseline adopts the 300-dimensional distributed embeddings [41] trained on 100 billion words from Google News.
  - GV + SVM: This baseline adopts the 300-dimensional Global Vectors [50] trained on 840 billion words from Common Crawl.
  - DS + SVM: This baseline employs the Skip-gram model [41] to train domain-specific word embeddings on each domain of the preprocessed reviews.

- Sentiment analysis also shows strengths in modeling helpfulness prediction. Following [11, 64], two fine-grained sentiment dictionaries are considered. SVM classifiers are then trained on extracted sentiment features.

  - LIWC + SVM: The Linguistic Inquiry and Word Count dictionary [49] presets 93 categories for contemporary English, including social and psychological states. The dictionary covers almost 6400 words, word stems, and emoticons.
  - GI + SVM: General Inquirer [59] attaches syntactic, semantic, and pragmatic information to part-of-speech tagged words. The dictionary contains 11,788 words assigned to 182 specified categories.

- RAT + SVM: This baseline trains linear SVM classifiers on the sole star rating information of reviews.
- CNN [26]: The vanilla CNN architecture for sentence classification.
- EG-CNN [3]: A variant of the vanilla CNN architecture where character embeddings and word-level embedding gates are used before convolution.
- CM1 [51]: A variant of the vanilla CNN architecture where raw rating values and content representations are concatenated.
- CM2 [51]: A variant of the vanilla CNN architecture where rating vectors and word embeddings are concatenated to learn content representations.
- MTNL [13]: A variant of the vanilla CNN architecture for multi-task learning, with character and word embeddings as inputs, attention on the convoluted feature maps, and raw rating regressing as the secondary task.

5.3 Hyperparameters and Training

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

Inside TRI, the content representation dimensionality is set to $m = d = 200$. The rating scale of $K = 5$ levels is adopted following Amazon and many contemporary e-commerce platforms. Rectified linear units are used for feature activation. Dropout operations of rate 0.5 are conducted on the penultimate layer to randomly mask half of the layer outputs. The remaining network weights are initialized using the Glorot uniform initializer [18]. Neural weights are updated through stochastic gradient descent over shuffled mini-batches using the mini-batch size of 64 and the Adam [28] update rule. Early stopping is performed when the validation loss has no improvement for 10 epochs.

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

For result reproducibility, all randomization processes involved in the paper are initialized with the same random seed. For result reliability, all neural models are trained and evaluated five times on each domain to report the average accuracy. SVM-based models are run once since the results are deterministic.

6 Result Analysis and Discussions

This section investigates TRI from several perspectives. Section 6.1 demonstrates the effectiveness of TRI. Section 6.2 performs ablation studies to validate individual TRI components. Section 6.3 compares the two rating enhancement methods TRI_{Add} and $\text{TRI}_{\text{Concat}}$, and discusses their performing behaviors. Finally, Sect. 6.4 provides qualitative analysis of the learned model weights (i.e., document embeddings, rating embeddings, and adaptive rating gates), followed by case studies.

6.1 Comparison with Baseline Methods

Table 4 reports the prediction accuracy of TRI against the baselines in helpfulness prediction. The bold results indicate models achieving the highest accuracy in each domain. TRI results higher than the baselines are in italics.

Table 4 Results of TRI against other methods

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

$*$ ($p < 0.1$), $**$ ($p < 0.05$), $***$ ($p < 0.01$)

In brief, TRI outperforms the baselines by approximately 1–5% in accuracy across domains. Both TFIDF + SVM and RAT + SVM set strong baselines for helpfulness prediction. The three types of pretrained word embeddings SGNS + SVM, GV + SVM, and DS + SVM achieve comparable performance to TFIDF + SVM, with far fewer dimensions at the price of about 1% loss in accuracy across domains. In particular, DS + SVM produces the highest performance, showing the necessity of using domain-specific word embeddings for neural model initialization. The two sentiment baselines LIWC + SVM and GI + SVM, however, are the worst among all baselines, suggesting that review sentiment alone may be insufficient for helpfulness learning.

The neural architectures except MTNL outweigh traditional ones in learning helpfulness information. CM2 on average achieves the closest performance to TRI. As will be discussed in Sect. 6.2, the effectiveness of CM2 is due to the vectorized encoding of rating information, which can be thought of as an implicit form of text–rating interaction. This again confirms that combining review content and ratings can assist in learning more expressive helpfulness information. Surprisingly, MTNL is worse than CNN and even traditional baselines in certain domains. The mediocre results require further investigation on the influence of review domains, data size, and model hyperparameters on model performance.

The effectiveness of TRI demonstrates the importance of the text–rating interaction. As discussed, review texts expresses the qualitative aspects of user opinions. The same opinion can also be measured quantitatively by the accompanying star rating. Whether the two perspectives are consistent can influence readers in perceiving review helpfulness. TRI aims at capturing such consistency during helpfulness modeling, which leads to improvement over the baselines. In the following subsections, the learned interactions will be discussed in further detail.

6.2 Ablation Studies

The following four TRI variants are considered to better understand the model behavior. Each variant disables a learning component of TRI to validate the change of model performance. Table 5 illustrates the accuracy of the four variants. Overall, TRI outperforms any of its variants, showing the necessity of the proposed TRI learning components to achieve the performance.

– $\text{TRI}_{\text{Plain}}$: The first variant uses only the content encoder. During model training, the adaptive learning gates in Eq. (8) are fixed to zero values $\boldsymbol{\sigma}_r = 0$ to exclude rating information. The learned content representations $\mathbf{h}$ are then used to predict review helpfulness.

Table 5 Performance of TRI variants

Variants	D1	D2	D3	D4	D5	D6
$\text{TRI}_{\text{Plain}}$	70.35	77.89	78.81	85.09	81.55	78.55
$\text{TRI}_{\text{Non-adaptive}}$ (Add)	71.97	78.23	80.04	86.80	82.90	80.00
$\text{TRI}_{\text{Non-adaptive}}$ (Concat)	72.33	78.83	79.86	86.89	82.93	79.92
$\text{TRI}_{\text{Raw-ratings}}$	70.36	77.38	78.79	85.12	81.43	78.75

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

Three comparison tasks are designed to further validate (1) the effectiveness of the review content encoder, (2) that of the gating mechanisms used for adaptive rating learning, and (3) that of the text–rating interaction. Table 6 summarizes the comparable items for each task.

6.2.1 Effectiveness of the Review Content Encoder

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

6.2.2 Effectiveness of the Gating Mechanisms

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

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

Table 6 Three comparison tasks

Task	Comparable items
Content encoder	$\text{TRI}_{\text{Plain}}$ against CNN and EG-CNN
Gating mechanisms	TRI_{Add} and $\text{TRI}_{\text{Concat}}$ against $\text{TRI}_{\text{Non-adaptive}}$ (Add) and $\text{TRI}_{\text{Non-adaptive}}$ (Concat)
Text–rating interaction	TRI_{Add}, $\text{TRI}_{\text{Concat}}$, $\text{TRI}_{\text{Non-adaptive}}$ (Add), and $\text{TRI}_{\text{Non-adaptive}}$ (Concat) against $\text{TRI}_{\text{Raw-ratings}}$

6.2.3 Effectiveness of the Text–Rating Interaction

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

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

6.3 Comparison Between the Combination Methods

Table 4 compares the performance between TRI_{Add} and $\text{TRI}_{\text{Concat}}$. As shown, one rating enhancement method does not consistently outperform another. The justification is the

emotional homogeneity between review texts and star ratings. Recall that the last fully connected layer in $\text{TRI}_{\text{Concat}}$ can be thought of as employing separate matrices to transform the learned content and rating representations. TRI_{Add} is a special case of $\text{TRI}_{\text{Concat}}$ in which the two matrices are shared, assuming higher homogeneity between the two sources. In domains where internal emotions in reviews texts are inadequate, $\text{TRI}_{\text{Concat}}$ may be more capable than TRI_{add} in performing text–rating interaction. Marco et al. [47] draw a similar conclusion that even using the same feature set can lead to domain-dependent performance. In their experiments on CD-related and movie-related reviews, the authors attribute similar performance to the two domains having more homogeneous products. In contrast, the electronic domain whose performance is far different includes many different types of products.

To further support the argument, the LIWC sentiment analysis is conducted to explore the emotional components of each domain. The analysis aims at showing the average percentage of words in reviews that possess either positive or negative emotions. As Table 7 reports, the three domains D1, D2, and D5, on which TRI_{Add} outperforms $\text{TRI}_{\text{Concat}}$, also possess higher ratios of emotional components. Given that the ratios are not proportional to the performance gains and threshold for emotion adequacy is unclear, the choice between TRI_{Add} and $\text{TRI}_{\text{Concat}}$ on new domains may require further domain-specific analysis. Nonetheless, TRI_{Add} is recommended since it entails less training parameters and yet yields similar performance.

6.4 Qualitative Analysis

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

6.4.1 Learned Document Embeddings

The first task illustrates the learned document embeddings used for helpfulness prediction. Specifically, the representations learned by TRI_{Add} and $\text{TRI}_{\text{Concat}}$ are compared against that by the TFIDF and CNN baselines. For each model, the output of the penultimate layer in Eqs. (9) and (10) is first computed. Dimensionality reduction via *t*-SNE [61] is then applied to obtain the two-dimensional vector representations.

Figure 7 presents the predicted document embeddings after training. As shown, review representations learned by the TFIDF + SVM baseline are mixed and the least separable. The vanilla CNN framework provides improved separability to distinguish one class from another. Still, there remain considerable overlaps between helpful and unhelpful reviews, in particular around the horizontal center. As for TRI_{Add} and $\text{TRI}_{\text{Concat}}$, different classes of reviews are further pushed to opposite directions and a clear boundary is observed, showing the effectiveness of TRI using text–rating interaction for helpfulness prediction.

6.4.2 Learned Rating Embeddings

The second task studies the learned rating embeddings in TRI_{Add} and $\text{TRI}_{\text{Concat}}$ to understand the mutual relationship among different star rating levels. Following text classification conventions, the closeness between two rating embeddings $\mathbf{e}'_{r1}$ and $\mathbf{e}'_{r2}$ is computed as their cosine similarity $\frac{\mathbf{e}'_{r1} \cdot \mathbf{e}'_{r2}}{\|\mathbf{e}'_{r1}\| \|\mathbf{e}'_{r2}\|} \in [-1, 1]$. The closer a returned score is to 1 (−1), the more similar (dissimilar) the two star levels are; 0 similarity indicates decorrelation.

Figure 8 illustrates the star rating similarity matrix, where the relationship between the five levels of star ratings is analyzed. Overall, the computed similarity values in TRI_{Add} and $\text{TRI}_{\text{Concat}}$ are both in compliance with the common understanding of star ratings. Take the one-star rating in TRI_{Add} as an example, its similarity with other ratings is inversely proportional to the star level. Also, a star level's previous and next neighbor possess closer similarity than the other levels. This shows that TRI can learn meaningful and effective rating embeddings.

The learned embeddings also reveal how customers perceive the meaning of star ratings. As discussed, star ratings quantitatively reflect customers' opinions and thus provide a reference sentiment for user satisfaction toward an item. While there is a consensus that one- and two-star (four- and five-star) ratings are perceived as negative (positive) experience, the perception of three-star reviews is usually ambiguous. As shown in the figure, the drastic drop in the similarity between three- and four-star rating clearly shows two polarity groups. The apparent division offers convincing evidence into rating-based review sentiment acquisition. Instead of

Table 7 Average ratio of emotional words across domains

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

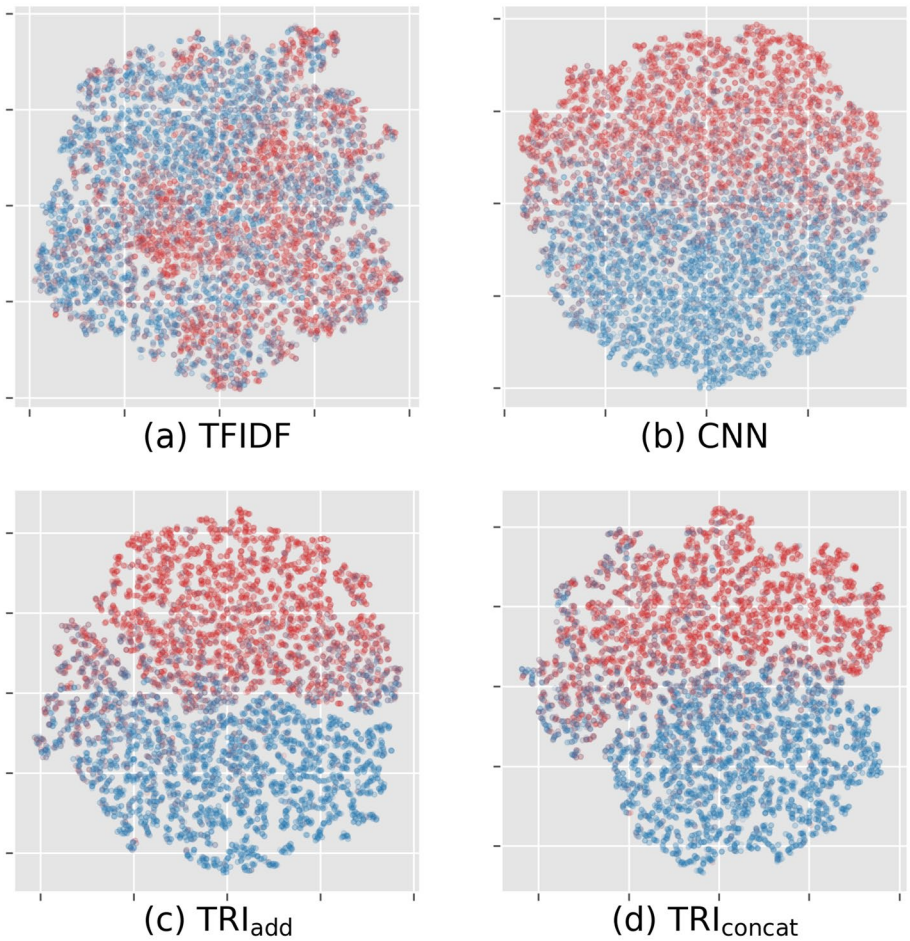

Fig. 7 *t*-SNE projection of the document embeddings learned by **a** the TFIDF + SVM baseline, **b** the vanilla CNN framework, **c** TRI_{Add}, and **d** TRI_{Concat}. *Blue and red points mark* helpful and unhelpful reviews, respectively

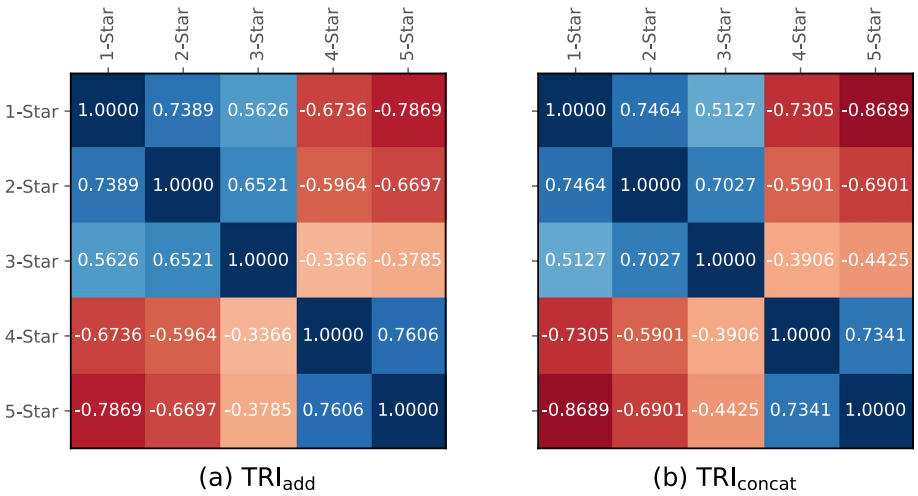

Fig. 8 Similarity between the learned rating embedding of individual star levels. *Blue (Red) color* indicates positive (negative) similarity

separating reviews into positive, neutral, and negative ones, dichotomization is a more realistic solution, with one-, two-, and three-star reviews being negative, and four- and five-star positive. The reason for three-star ratings being treated as a negative emotion is highly related to the online social context. As pointed out by [34], customers tend to provide positive feedback, which diminishes the neutrality of three-star ratings.

The aforementioned findings can hopefully inspire improvement on the Likert-based rating systems used for

quantifying customer satisfaction. Since customers tend to express opinions dichotomously, adjustment can be made to emphasize the positivity and negativity of customer opinions. For instance, four-point Likert scales or Yes/No questions.

6.4.3 Learned Adaptive Rating Ratios

The third task investigates the dependence of review content on rating information. Figure 9 plots the gates $\boldsymbol{\sigma}_r$ in Eq. (8) learned by TRI_{Add}. The results of $\text{TRI}_{\text{Concat}}$ are similar to TRI_{Add} and thus skipped. Due to limited space, only the first 60 helpful and unhelpful samples in the testing

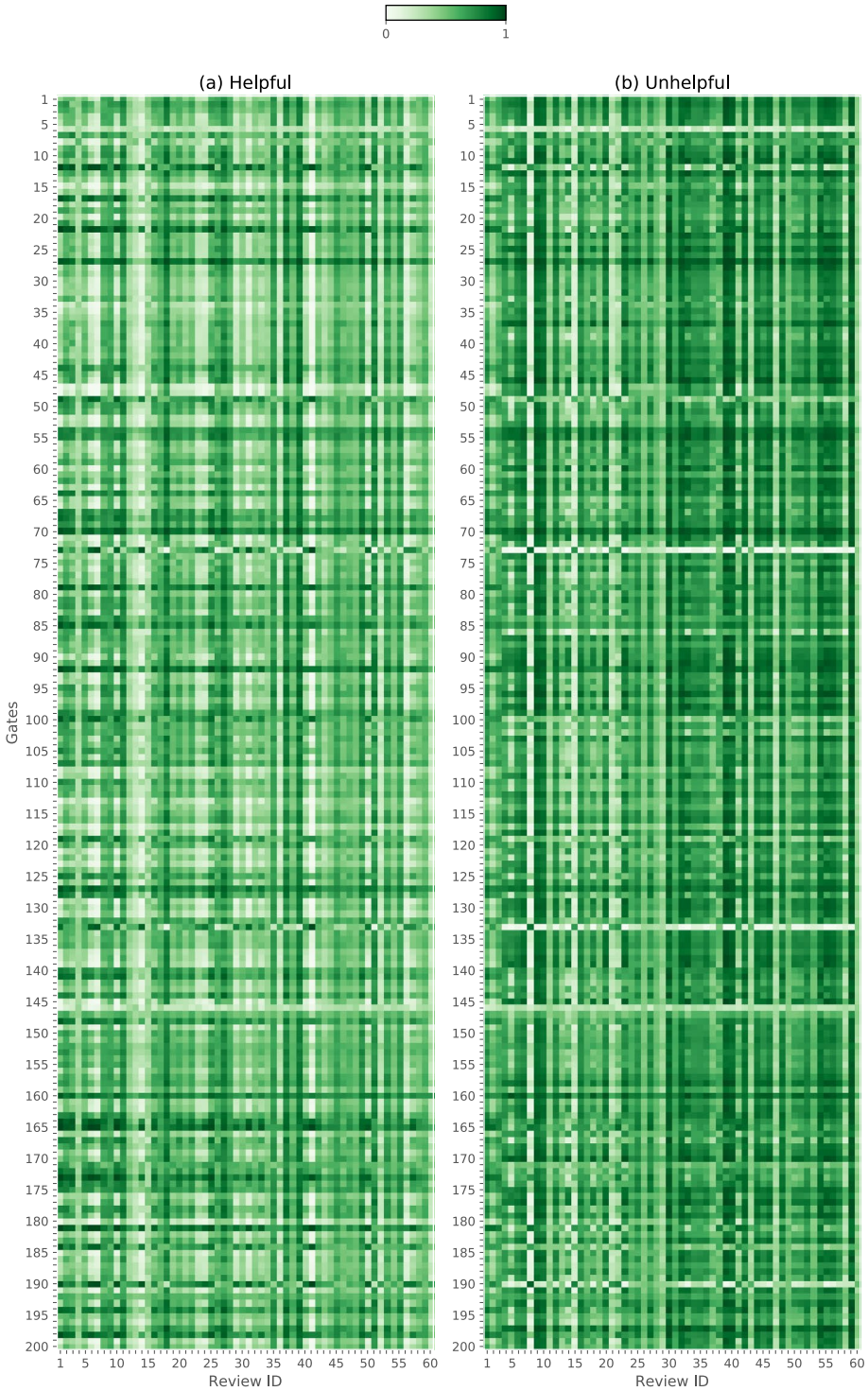

Fig. 9 Learned amount of rating information required by texts in **a** helpful and **b** unhelpful reviews

Table 8 Examples of real-world reviews influenced by their star ratings

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

From left to right, each triplet indicates (1) the text-only helpfulness predicted by CNN, (2) the text–rating interactive helpfulness predicted by TRI_{Add}, and (3) the ground truth

set are demonstrated. Each column consists of 200 adaptively learned ratios, respectively, determining the amount of rating information needed by a review's learned content representation. The ratios ranging from 0 to 1 indicate the importance of individual rating embedding dimensions.

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

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

6.4.4 Case Studies

In the fourth task, the effectiveness of TRI is demonstrated with real-world examples. Table 8 showcases four reviews randomly chosen from the test set. The CNN baseline is used for non-rating helpfulness prediction, whereas TRI_{Add} is used for establishing the text–rating interaction. In review (a), the author appreciated the CD product overall, but was dissatisfied with the price. Since readers did not expect such a comment would lead to a one-star rating, the contrast makes the review less helpful. Similarly, the mismatch between the text and rated star in review (b) confuses helpfulness perception. Review (c) marks an opposite situation where relatively negative comments were rated four stars, weakening the convincing power. Although review (d) mostly expressed negative opinions, the author suggested that the disappointment is rather regretful feelings than dissatisfaction. The four-star rating further validates and reinforces the impression, which brings high trustworthiness. The aforementioned samples provide strong evidence that text–rating interaction plays an important role in the perceptual process of review helpfulness.

7 Conclusion and Future Work

This paper has presented TRI, a deep neural architecture that learns the interaction between review texts and star ratings for helpfulness prediction. In contrast to prior

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

From a practical perspective, TRI can be hopefully integrated into existing helpfulness prediction systems. TRI takes as input review texts and star ratings for helpfulness modeling; both are standard components of a review on nearly all contemporary e-commerce platforms. Two common integration methods are available. When TRI is used as a means for feature representation, the document embedding of a review learned by TRI can be used to complement that learned by an existing system. The two sets of features are then combined, upon which classification/regression algorithms are applied for final helpfulness prediction. Alternatively, TRI can be regarded as another base estimator in addition to the existing system. The final helpfulness of a review will be determined based on the predicted labels from the two (or even more) models, using max voting, weighted average, or more advanced ensemble learning techniques.

There remain several directions to be addressed. (1) Further sensitivity analysis of the TRI hyperparameters will be conducted to investigate model performance, in particular the dimensionality of word vectors, content representations, and rating embeddings. (2) Frequent text–rating interaction patterns and domain-specific characteristics will be summarized from the trained models. Further investigation on the gating behaviors and the discrepancy (if any) in text–rating interaction between domains can hopefully offer more insights. (3) More advanced approaches will be developed to further address the interpretation of individual rating embedding dimensions and their relationship with review texts. (4) The interaction between review texts and star ratings will be constructed using more sophisticated structures such as attention mechanisms or sentence-level rating information. The diversity between reviewers in giving star ratings will also be considered. (5) The extent to which existing review characteristics (e.g., review length, text valence) affect the text–rating interaction will be studied. The characteristics will also be included in TRI for multi-characteristic interaction. (6) Inspired by existing studies working on transfer learning, the learned interactive knowledge from one domain will be applied to another. It is also interesting to build an integrated model for multi-domain helpfulness prediction.

Author Contributions Jiahua Du conceived and designed the work, obtained, analyzed, and interpreted the data, created the new software used in the work, made the initial draft of the work, and revised the work. Liping Zheng obtained and interpreted the data, and created the new software used in the work. Jiantao He analyzed and interpreted the data, and created the new software used in the work. Jia Rong, Hua Wang, and Yanchun Zhang revised the work.

Funding The authors received no specific funding for this work.

Availability of data and materials The datasets generated and/or analyzed during the current study will be available on https://github.com/tokawah/Helpfulness-Measurement.

Compliance with Ethical Standards

Competing Interests The authors declare that they have no competing interests.

References

1. Baek H, Lee S, Oh S, Ahn JH (2015) Normative social influence and online review helpfulness: polynomial modeling and response surface analysis. J Electron Commer Res 16:290–306
2. Charrada EB (2016) Which one to read? Factors influencing the usefulness of online reviews for RE. In: 2016 IEEE 24th international requirements engineering conference workshops (REW), pp 46–52
3. Chen C, Qiu M, Yang Y, Zhou J, Huang J, Li X, Bao FS (2019) Multi-domain gated CNN for review helpfulness prediction. In: The world wide web conference, WWW '19. ACM, New York, NY, USA, pp 2630–2636
4. Chen C, Yang Y, Zhou J, Li X, Bao FS (2018) 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). Association for Computational Linguistics, New Orleans, Louisiana, pp 602–607
5. Chen C, Zhang M, Liu Y, Ma S (2018) Neural attentional rating regression with review-level explanations. In: Proceedings of the 2018 world wide web conference, WWW '18. International

World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, pp 1583–1592
6. Chiriatti G, Brunato D, Dell'Orletta F, Venturi G (2019) What makes a review helpful? predicting the helpfulness of Italian Tripadvisor reviews. In: Raffaella B, Roberto N, Giovanni S (eds) Italian Conference on Computational Linguistics (CLiC-it). CEUR, Bari, Italy
7. Cho K, Van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y (2014) Learning phrase representations using RNN encoder–decoder for statistical machine translation. ArXiv preprint arXiv:1406.1078
8. Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, Kuksa P (2011) Natural language processing (almost) from scratch. J Mach Learn Res 12:2493–2537
9. Danescu-Niculescu-Mizil C, Kossinets G, Kleinberg J, Lee L (2009) How opinions are received by online communities: a case study on amazon.com helpfulness votes. In: Proceedings of the 18th international conference on world wide web, WWW '09. ACM, New York, NY, USA, pp 141–150
10. Dauphin YN, Fan A, Auli M, Grangier D (2017) Language modeling with gated convolutional networks. In: Proceedings of the 34th international conference on machine learning, volume 70. JMLR.org, pp 933–941
11. Du J, Rong J, Michalska S, Wang H, Zhang Y (2019) Feature selection for helpfulness prediction of online product reviews: an empirical study. PLOS One 14(12):1–26
12. Du J, Rong J, Wang H, Zhang Y (2019) Helpfulness prediction for online reviews with explicit content–rating interaction. In: Web information systems engineering (WISE). Springer International Publishing, Hong Kong SAR, China, pp 795–809
13. Fan M, Feng Y, Sun M, Li P, Wang H, Wang J (2018) 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), pp 343–350
14. Fan M, Feng C, Guo L, Sun M, Li P (2019) Product-aware helpfulness prediction of online reviews. In: The world wide web conference, WWW '19. ACM, New York, NY, USA, pp 2715–2721
15. Fang B, Ye Q, Kucukusta D, Law R (2016) Analysis of the perceived value of online tourism reviews: influence of readability and reviewer characteristics. Tour Manage 52:498–506
16. Ge S, Qi T, Wu C, Wu F, Xie X, Huang Y (2019) Helpfulness-aware review based neural recommendation. CCF transactions on pervasive computing and interaction
17. Ghose A, Ipeirotis PG (2011) Estimating the helpfulness and economic impact of product reviews: mining text and reviewer characteristics. IEEE Trans Knowl Data Eng 23(10):1498–1512
18. Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics, pp 249–256
19. Haque ME, Tozal ME, Islam A (2018) Helpfulness prediction of online product reviews. In: Proceedings of the ACM symposium on document engineering (2018) DocEng '18. ACM, New York, NY, USA, pp 35:1–35:4
20. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR), pp 770–778
21. He R, McAuley J (2016) Ups and downs: modeling the visual evolution of fashion trends with one-class collaborative filtering. In: Proceedings of the 25th international conference on world wide web, WWW '16. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, pp 507–517
22. Hoffait A-S, Ittoo A, Schyns M (2018) Assessing and predicting review helpfulness: critical review, open challenges and research agenda. In: 29ème conférence européenne sur la recherche opérationnelle (EURO2018)
23. Huang AH, Chen K, Yen DC, Tran TP (2015) A study of factors that contribute to online review helpfulness. Comput Hum Behav 48:17–27
24. Kalchbrenner N, Grefenstette E, Blunsom P (2014) A convolutional neural network for modelling sentences. In: Proceedings of the 52nd annual meeting of the association for computational linguistics (volume 1: long papers). Association for Computational Linguistics, Baltimore, Maryland, pp 655–665
25. Kim S-M, Pantel P, Chklovski T, Pennacchiotti M (2006) Automatically assessing review helpfulness. In: Proceedings of the 2006 conference on empirical methods in natural language processing, EMNLP '06. Association for Computational Linguistics, Stroudsburg, PA, USA, pp 423–430
26. Kim Y (2014) Convolutional neural networks for sentence classification. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). Association for Computational Linguistics, Doha, Qatar, pp 1746–1751
27. Kim Y, Jernite Y, Sontag D, Rush AM (2016) Character-aware neural language models. In: Proceedings of the thirtieth AAAI conference on artificial intelligence, AAAI'16. AAAI Press, pp 2741–2749
28. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. CoRR, arXiv:abs/1412.6980
29. Korfiatis N, García-Bariocanal E, Sánchez-Alonso S (2012) Evaluating content quality and helpfulness of online product reviews: the interplay of review helpfulness vs. review content. Electron Commer Res Appl 11(3):205–217
30. Kozinets RV (2016) Amazonian forests and trees: multiplicity and objectivity in studies of online consumer-generated ratings and reviews, a commentary on De Langhe, Fernbach, and Lichtenstein. J Consum Res 42(6):834–839
31. Krishnamoorthy S (2015) Linguistic features for review helpfulness prediction. Expert Syst Appl 42(7):3751–3759
32. Lee S, Choeh JY (2014) Predicting the helpfulness of online reviews using multilayer perceptron neural networks. Expert Syst Appl 41(6):3041–3046
33. Lipton ZC, Berkowitz J, Elkan C (2015) A critical review of recurrent neural networks for sequence learning. ArXiv preprint arXiv:1506.00019
34. Liu J, Cao Y, Lin C-Y, Huang Y, Zhou M (2007) Low-quality product review detection in opinion summarization. In: Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL). Association for Computational Linguistics, Prague, Czech Republic, pp 334–342
35. Lu Y, Tsaparas P, Ntoulas A, Polanyi L (2010) Exploiting social context for review quality prediction. In: Proceedings of the 19th international conference on world wide web, WWW '10. ACM New York, NY, USA, pp 691–700
36. Ma Y, Xiang Z, Qianzhou D, Fan W (2018) Effects of user-provided photos on hotel review helpfulness: an analytical approach with deep leaning. Int J Hosp Manage 71:120–131
37. Malik MSI, Hussain A (2017) Helpfulness of product reviews as a function of discrete positive and negative emotions. Comput Hum Behav 73:290–302
38. Martin L, Pu P (2014) Prediction of helpful reviews using emotions extraction. In: Proceedings of the twenty-eighth AAAI conference on artificial intelligence, AAAI'14. AAAI Press, pp 1551–1557
39. Melián-González S, Bulchand-Gidumal J, López-Valcárcel BG (2013) Online customer reviews of hotels: as participation increases, better evaluation is obtained. Cornell Hosp Q 54(3):274–283

40. Matthias M, Nikolaos K, Zicari RV (2014) Using dependency bigrams and discourse connectives for predicting the helpfulness of online reviews. In: Hepp M, Hoffner Y (eds) E-commerce and web technologies. Springer, Cham, pp 146–152
41. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems, pp 3111–3119
42. Mudambi SM, Schuff D (2010) Research note: What makes a helpful online review? a study of customer reviews on amazon. com. MIS Q 34(1):185–200
43. Murphy R (2018) Local consumer review survey. https://www.brightlocal.com/research/local-consumer-review-survey/
44. Ocampo DG, Ng V (2018) Modeling and prediction of online product review helpfulness: a survey. In: Proceedings of the 56th annual meeting of the association for computational linguistics (volume 1: long papers). Association for Computational Linguistics, Melbourne, Australia, pp 698–708
45. Park S, Nicolau JL (2015) Asymmetric effects of online consumer reviews. Ann Tour Res 50:67–83
46. Park Y-J (2018) Predicting the helpfulness of online customer reviews across different product types. Sustainability 10(6):1735
47. Passon M, Lippi M, Serra G, Tasso C (2018) Predicting the usefulness of Amazon reviews using off-the-shelf argumentation mining. In: Proceedings of the 5th workshop on argument mining. Association for Computational Linguistics, Brussels, Belgium, pp 35–39
48. Paul D, Sarkar S, Chelliah M, Kalyan C, Nadkarni PPS (2017) Recommendation of high quality representative reviews in e-commerce. In: Proceedings of the eleventh ACM conference on recommender systems, RecSys '17. ACM, New York, NY, USA, pp 311–315
49. Pennebaker JW, Boyd RL, Jordan K, Blackburn K (2015) The development and psychometric properties of LIWC2015. Technical report, The University of Texas at Austin
50. Pennington J, Socher R, Manning CD (2014) Glove: global vectors for word representation. In: Empirical methods in natural language processing (EMNLP), pp 1532–1543
51. Qu X, Li X, Rose JR (2018) Review helpfulness assessment based on convolutional neural network. CoRR, arXiv:abs/1808.09016
52. Quaschning S, Pandelaere M, Vermeir I (2015) When consistency matters: the effect of valence consistency on review helpfulness. J Comput Mediat Commun 20(2):136–152
53. Roy G, Datta B, Mukherjee S (2018) Role of electronic word-of-mouth content and valence in influencing online purchase behavior. J Mark Commun 25:661–684
54. Salehan M, Kim DJ (2016) Predicting the performance of online consumer reviews: a sentiment mining approach to big data analytics. Decis Support Syst 81:30–40
55. Saumya S, Singh JP, Dwivedi YK (2019) Predicting the helpfulness score of online reviews using convolutional neural network. Soft Comput. https://doi.org/10.1007/s00500-019-03851-5
56. Schuckert M, Liu X, Law R (2016) Insights into suspicious online ratings: direct evidence from TripAdvisor. Asia Pac J Tour Res 21(3):259–272
57. Shin S, Chung N, Xiang Z, Koo C (2019) Assessing the impact of textual content concreteness on helpfulness in online travel reviews. J Travel Res 58(4):579–593
58. Siering M, Muntermann J, Rajagopalan B (2018) Explaining and predicting online review helpfulness: the role of content and reviewer-related signals. Dec Support Syst 108:1–12
59. Stone PJ, Bales RF, Namenwirth JZ, Ogilvie DM (1966) The general inquirer: a computer system for content analysis and retrieval based on the sentence as a unit of information. Behav Sci 7(4):484–498
60. Tsang ASL, Prendergast G (2009) Is a "star" worth a thousand words? The interplay between product-review texts and rating valences. Eur J Mark 43(11/12):1269–1280
61. van der Maaten L, Hinton G (2008) Visualizing data using t-SNE. J Mach Learn Res 9:2579–2605
62. Xiang Z, Qianzhou D, Ma Y, Fan W (2017) A comparative analysis of major online review platforms: implications for social media analytics in hospitality and tourism. Tour Manage 58:51–65
63. Yang Y, Chen C, Bao FS (2016) Aspect-based helpfulness prediction for online product reviews. In: 2016 IEEE 28th international conference on tools with artificial intelligence (ICTAI), pp 836–843
64. Yang Y, Yan Y, Qiu M, Bao F (2015) Semantic analysis and helpfulness prediction of text for online product reviews. In: Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 2: short papers). Association for Computational Linguistics, Beijing, China, pp 38–44
65. Yin D, Bond SD, Zhang H (2014) Anxious or angry? Effects of discrete emotions on the perceived helpfulness of online reviews. MIS Q 38(2):539–560
66. Yin D, Mitra S, Zhang H (2016) Research note—when do consumers value positive vs. negative reviews? An empirical investigation of confirmation bias in online word of mouth. Inf Syst Res 27(1):131–144
67. Zhang R, Tran T, Mao Y (2012) Opinion helpfulness prediction in the presence of "words of few mouths". World Wide Web 15(2):117–138
68. Zhou S, Guo B (2015) The interactive effect of review rating and text sentiment on review helpfulness. In: Stuckenschmidt H, Jannach D (eds) E-commerce and web technologies. Springer, Cham, pp 100–111