Abstract

Helpful reviews have been essential for the success of e-commerce services, as they help customers make quick purchase decisions and benefit the merchants in their sales. While many reviews are informative, others provide little value and may contain spam, excessive appraisal, or unexpected biases. With the large volume of reviews and their uneven quality, the problem of detecting helpful reviews has drawn much attention lately. Existing methods for identifying helpful reviews primarily focus on review text and ignore the two key factors of (1) who post the reviews and (2) when the reviews are posted. Moreover, the helpfulness votes suffer from scarcity for less popular products and recently submitted (a.k.a., cold-start) reviews. To address these challenges, we introduce a dataset and develop a model that integrates the reviewer’s expertise, derived from the past review history of the reviewers, and the temporal dynamics of the reviews to automatically assess review helpfulness. We conduct experiments on our dataset to demonstrate the effectiveness of incorporating these factors and report improved results compared to several well-established baselines.

1 Introduction

Many customers rely on online reviews from non-professionals, on daily basis, to decide what products to buy (e.g., Amazon), what hotels to stay at (e.g., TripAdvisor), what restaurants to eat (e.g., Yelp) and even what books to read (e.g., Goodreads). A recent survey of Bizrate Insights reward members found that approximately 98% of online shoppers research a vendor via online reviews before making a purchase decision (Kats, 2018). Since the reviews are expected to describe the actual experiences and opinions of users, they can provide a reliable source of reference, improving other customers’ confidence, comfort, and the overall shopping experience (Foo et al., 2017; Gamzu et al., 2021). However, despite their tremendous benefits, online reviews are often of mixed qualities. While many reviews are informative, others provide little value and may contain excessive appraisal or spam (see Figure 1-c). There are multiple factors that affect the quality of a review, including the reviewers’ life experience, educational background, and the motive for writing the review (Du et al., 2021), and these factors are not usually explicit in the review text. All these pose challenges for customers who are less experienced in a subject area and need the reviews the most, simply because there is less incentive for more experienced users to use the reviews. Moreover, customers usually have limited patience for reading reviews – most customers read less than 10 reviews before mak-
ing a purchase decision about an item (Murphy, 2016). The large volume of reviews and their un-
predictable quality and the limited customer pa-
tience demand better review utilization strategies
to manage the information overload.

One standard method to identify more informa-
tive reviews is to ask for feedback from customers
or site visitors who read them. By asking, “Was
this review helpful to you?”, or “Did you find this
review helpful?” at the end of each review, online
platforms can crowdsourcing helpfulness votes from
other customers. As a result, user reviews that gain
the most helpful votes are shown first to the poten-
tial buyers to make the decision easier. However,
the voting data suffers from scarcity (Siersdorfer
et al., 2010) since only a tiny proportion of cus-
tomers are willing to cast helpfulness votes. The
scarcity is even more severe in reviews of less pop-
ular products and more recently submitted reviews
(a.k.a., cold-start reviews) (Liu et al., 2008), despite
the fact that more recent reviews may in fact con-
tain more relevant and time-sensitive information
(e.g., “New COVID Restrictions” or “Dirty Pool
Area”) as shown in Figure 1-b but no helpfulness
vote.

In this paper, we study the confluence of the re-
viewing history of reviewers and the review text
for helpfulness identification. First, we observe
that people who post more reviews and earn more
helpful votes are more likely to be better reviewers.
Second, trustworthy reviewers (e.g., Figure 1-a) are
less likely to be posting fake or biased reviews, and
their reviews are more likely to earn more helpful
votes; otherwise, they will be ruining their reputa-
tion. Third, those who have been to more hotels
or restaurants across different cities have a better
basis for comparison and writing critical reviews.
To the best of our knowledge, existing works only
focus on review content and neglect the reviewers
and their reviewing history. Integrating the review
text with the reviewing history of the reviewers is
the problem studied in this paper.

Our main contributions are summarized as fol-

• We introduce a new dataset with both review
text and reviewer’s history, to highlight the
importance of integrating the two sources for
review helpfulness.

• We propose a model incorporating the re-
viewer’s expertise and temporal information
of reviews in helpfulness prediction.

• We present a detailed case-study to interpret
the model behavior and highlight potential
directions to be addressed in the future.

2 Related work

More traditional approaches on review helpfulness
prediction focus solely on the text of reviews, and
some consider both text and images to guide the
prediction. In general, the task can be addressed
using a predictive model based on hand-crafted fea-
tures such as structural (Susan and David, 2010;
Xiong and Litman, 2014), lexical (Kim et al.,
2006; Xiong and Litman, 2011), syntactic (Kim
et al., 2006), emotional (Martin and Pu, 2014),
semantic (Yang et al., 2015), and arguments (Liu
et al., 2017) from the review text. These features
may be fed into a conventional classifier such as
SVM, Random Forest, or gradient boosting to iden-
tify helpful reviews. These methods heavily rely
on manual feature engineering, which is labor-
intensive and time-consuming.

Inspired by the remarkable progress of deep neu-
ral networks, more recent studies make use of deep
neural models, which can learn both intrinsic and
extrinsic features given labeled data. Chen et al.
(2018) uses a text-based CNN model to automati-
cally capture the character-level, word-level, and
topic-level features for helpfulness prediction. Fan
et al. (2018) uses an end-to-end multi-task neural
architecture with the help of an auxiliary task, such
as rating regression, to boost the performance of the
review helpfulness identification. Liu et al. (2021)
and Han et al. (2022) use both text and images to
guide the review helpfulness prediction. Since the
image field is usually optional in reviews, a large
volume of reviews contain only text, for which
these multimodal models would produce inconsis-
tent results.

3 Review Helpfulness Prediction

3.1 Dataset

To the best of our knowledge, there is no human-
annotated dataset that is publicly available for the
task of review helpfulness prediction with the re-
viewers’ attributes and review date. Therefore,
we build our dataset by scraping reviews from
TripAdvisor1. Out of 225,664 reviews retrieved,
close to one third have no helpful votes. We filter
such reviews, and this reduces the number of re-
views to 161,541. Table 1 presents the summary of

1https://www.tripadvisor.com
our dataset with train, validation, and test splits. Following (Liu et al., 2021), we leverage a logarithmic scale (\(\log_2 n_{\text{votes}}\)) to categorize the reviews based on the number of votes received. Specifically, we map the number of votes into five intervals (i.e., \([1,2), [2,4), [4,8), [8,16), [16,\infty)\)), each corresponding to a helpfulness score \(Y \in \{1,2,3,4,5\}\), where the higher the score, the more helpful the review.

### 3.2 Proposed Model

**Review Helpfulness Prediction (RHP)** can be modeled as a supervised machine learning task where the input contains information about the reviews (\(R\)) and the reviewers (\(U\)). Let \(R_i = ([s_1,\ldots,s_N], t_i)\) denote a review posted at time \(t_i\) with sentences \(s_1,\ldots,s_N\), and \(U_i = (n_i, m_i)\) denote a reviewer who posts \(n_i\) reviews and earns a total of \(m_i\) helpful votes. We formulate the review helpfulness prediction as a multi-class classification where we seek to find a model \(f\) that minimizes the loss function \(L\), i.e.

\[
\min_\theta L (f(\theta, R, U), Y),
\]

where \(Y\) is the ground-truth, \(\theta\) is the model parameter and the output of the model is a helpfulness class \(Y \in \{1,2,3,4,5\}\). The learning task is to find the best parameter that minimizes the above equation.

We encode the review sentences using BERT (Devlin et al., 2019; Xu et al., 2019). We concatenate the review sentences together while inserting a [CLS] token at the start and a [SEP] token at the end. If \(h_{[\text{CLS}]}^{(i)}\) denotes the embedding vector of the special [CLS] token and \(h_{[\text{SEP}]}^{(i)}\) denotes the embedding vector of the \(i\)-th token, we extract the last hidden state of \(h_{[\text{CLS}]}^{(i)}\) to represent the review sentences and apply a linear transformation to get a final contextualized representation \(x_h \in \mathbb{R}^K\), where \(\Theta\) is a non-linear activation function.

\[
[h_{[\text{CLS}]}^{(i)}, h_{[\text{SEP}]}^{(i)}, \ldots] = \text{BERT}([\text{CLS}] s_1,\ldots,s_N [\text{SEP}]),
\]

\[
x_h = \Theta (\text{MLP} (h_{[\text{CLS}]}^{(i)})).
\]

Generally, users who post more reviews and earn more helpful votes are likely to be better reviewers. Such users may have been to more hotels and restaurants across the globe and have a better basis for comparison. We define the term reviewer expertise as the mean number of helpful votes received per review, written as \(e_s = m/n\) for a reviewer who posts \(m\) reviews and earns \(n\) overall helpfulness votes. We use a linear layer to get a weighted representation of the expertise score \(h_s\).

\[
h_s = \text{MLP} (e_s)
\]

Previous approaches for this task fail to consider the temporal nature of the reviews. Older reviews are more likely to accumulate more helpfulness votes than newer reviews but are not necessarily the most relevant describing the current conditions (e.g., new COVID restrictions). One-time problems such as broken bathrooms and dirty pool area are likely to be addressed and to be less relevant. Let \(t_d\) be the relative age of a review in days, for example, as of the day the reviews are scraped. We use a linear layer to get a weighted representation of the relative review age.

\[
h_t = \text{MLP} (t_d).
\]

It should be noted that both the review age and the reviewer expertise are normalized to a fixed range \([a,b]\) before being used in the linear layers in Equations 4 and 5. If \(X\) denotes a set of scores (e.g., reviewers expertise score), a score \(x_i \in X\) is normalized into \(z_i\) as follows:

\[
z_i = (b-a) \frac{x_i - \min(X)}{\max(X) - \min(X)} + a
\]

In our case, both review age and reviewer expertise are scaled into the interval \([0,1]\).

We concatenate the textual representation \((x_h)\), expertise representation \((h_s)\), and temporal representation \((h_t)\) to get a final embedding

\[
o_{\text{final}} = h_s \oplus x_h \oplus h_t,
\]

where \(\oplus\) is a concatenation operator. The final helpfulness prediction layer feeds \(o_{\text{final}}\) into a linear layer and use softmax activation to get the final predicted helpfulness class \(\hat{Y}\).

\[
\hat{Y} = \text{softmax} (W_r \cdot o_{\text{final}} + b_r),
\]
We compare our system with well-established baselines. As part of a detailed evaluation of our algorithm, we conduct detailed ablation studies to demonstrate the effects of different components of our RHP model by removing expertise (denoted as w/o Expertise) and removing temporal information (denoted as w/o Temporal). The ablation test results on our dataset are summarized in Table 2. We can observe that the temporal feature has the largest impact on the performance of our model, and the impact of expertise is also significant. This suggests that the reviewer’s expertise and temporal information of the reviews play a key role in review helpfulness prediction. Therefore, it is no surprise that combining all components achieves the best performance on our proposed dataset.

### 3.3.2 Analysis

We also present a detailed analysis to provide more supportive evidence of our arguments. To this end, we randomly selected \( m \) examples for each class of reviews considering helpfulness votes. Then, we extract Top \( K \) (where \( K = 5 \)) \( n \)-grams from each class of reviews to identify the most relevant keywords or topics in reviews to assess what aspects are most talked about the items (e.g., hotels or restaurants).

#### Preprocessing

Our preprocessing step includes tokenization, lemmatization, removal of stopwords, Part-Of-Speech (POS) tagging, and filtering punctuation marks. We use the NLTK\(^4\) to preprocess each sentence and obtain a more accurate representation of the information. Moreover, we also add ‘hotel’ and ‘restaurant’ in the stopwords list as they frequently occur in every review and are not meaningful in our context.

#### Extracting Candidate \( n \)-grams

We remove the sentiment words and emojis using VADER\(^5\) (Hutto and Gilbert, 2014), a "gold-standard" sentiment lexicon especially attuned to microblog-like contexts. As the sentiment expressed in reviews are highly subjective, we are interested in extracting only the aspects or topics (e.g., room, location, customer service etc.) for which the opinions are

---

\(^3\)Code, dataset, and model checkpoints: https://github.com/tafeer-nayeem/RHP

\(^4\)https://www.nltk.org/

\(^5\)https://github.com/cjhutto/vaderSentiment

---

Table 2: Performance compared to our baseline models and the result of our ablation study (↑ indicates higher values for a better performance and ↓ indicates lower values for a better performance). \(^\dagger\) reported results are statistically significant in paired t-test by taking \( p \)-value < 0.05.

\[
\mathcal{L} = \mathcal{L}_{CE}(\hat{Y}, Y) \tag{9}
\]

where \( W_r \in \mathbb{R}^{K \times K} \) and \( b_r \in \mathbb{R}^K \) denote the projection parameter and a bias term respectively. We use the cross-entropy loss function \( \mathcal{L}_{CE} \) with respect to the ground truths helpfulness class (\( Y \)).

### 3.3 Experiments

We evaluated the performance of the proposed model\(^3\) compared to well-established baselines. We compare our system with ARH (Kim et al., 2006), UGR + BGR (Xiong and Litman, 2011), TextCNN (Chen et al., 2018), MTNL (Fan et al., 2018), and BERTHelp (Xu et al., 2020). We didn’t perform any explicit preprocessing of the review text. We discuss the baseline systems, preprocessing, and hyperparameters used for our experiments in Appendix (Section B & Section C).

#### 3.3.1 Results

As part of a detailed evaluation of our algorithm, we report our model’s performance compared with the baselines in terms of Accuracy (Acc.), Mean Average Error (MAE), and Mean Squared Error (MSE). As shown in Table 2, our final model outperforms the baselines in terms of all the metrics. Our ground-truth values consist of 5 classes which correspond to five helpfulness scores \{1, 2, 3, 4, 5\}, where the higher the score, the more helpful the review. To gain more insights into the performance of our prediction model, we also evaluate our algorithm in terms of MAE and MSE, which assess the fine-grained differences between the ground-truth and the predicted helpfulness scores. Our RHP model consistently outperforms the baselines with a good margin, which means when misclassified, our model predictions are very close to the actual helpfulness scores. We conduct detailed ablation studies to demonstrate the effects of different components of our RHP model by removing expertise (denoted as w/o Expertise) and removing temporal information (denoted as w/o Temporal). The ablation test results on our dataset are summarized in Table 2. We can observe that the temporal feature has the largest impact on the performance of our model, and the impact of expertise is also significant. This suggests that the reviewer’s expertise and temporal information of the reviews play a key role in review helpfulness prediction. Therefore, it is no surprise that combining all components achieves the best performance on our proposed dataset.

---

\(^3\)Code, dataset, and model checkpoints: https://github.com/tafeer-nayeem/RHP

\(^4\)https://www.nltk.org/

\(^5\)https://github.com/cjhutto/vaderSentiment
Table 3: Top 5 unigrams and bigrams extracted from five different classes of reviews divided according to helpfulness votes. For each column, green color indicates the overlap with all 5 classes, whereas blue for 4, orange for 3, and red for 2 overlaps.

expressed. Therefore, we keep only the nouns\(^6\) (with POS tags ‘NN’ and ‘NNS’) for extracting the aspects or topics.

### Ranking Candidate \(n\)-grams

We extract the unigrams and bigram collocations for each of the review classes. Then, we rank the unigrams by counting the frequency of occurrences and bigrams using likelihood ratios (Manning and Schutze, 1999) to obtain Top \(K\). We present the Top 5 unigrams and bigrams in Table 3 grouped according to helpfulness classes and ordered by descending ranking scores.

Table 3 shows a high overlap of \(n\)-grams among different classes of reviews, which further strengthen our argument that helpfulness does not entirely depend on the review text but rather the confluence of the review text, reviewing history of reviewers (who post the reviews), review age (when the reviews are posted). Generally, older reviews (i.e., review age) were present longer than the newer reviews in the platform and had more time to accumulate helpful votes.

---

\(\)

\(^6\)As adjectives and adverbs may contain sentiment towards aspects.

---

Figure 2: Top 10 ranked tokens of the RHP model shown in green colors with the color intensity indicating the importance of the tokens in the overall prediction.

### 3.4 Case Study

To gain more insights into the review helpfulness prediction task, we present a detailed case-study to interpret the model behavior and highlight the most important features of this task. Models are interpretable when humans can readily comprehend the reasoning behind model predictions and decisions made (Kim et al., 2016). To this end, we randomly selected a sample with Helpfulness Class \(= 3\) from our test set and used Captum\(^7\) to interpret the words/tokens that contributed the most to the prediction. As can be seen in Figure 2, the top-ranked words are highly representative of the aspects/facilities listed on the restaurant page. We can conclude from this observation that users tend to look for specific aspects in reviews to find them helpful. We also notice that the use of personal pronouns (e.g., I, we, they, etc.), describing personal experiences, contributes to the helpfulness prediction. People often find reviews useful if it comes from others’ experiences and personal pronouns are a good indicator of it.

---

\(^7\)Captum (https://captum.ai/) is an open-source, extensible library for model interpretability that uses the integrated gradients method (Sundararajan et al., 2017).

---

4 Conclusion and Future Work

In this paper, we develop a model incorporating the reviewer’s expertise and temporal information in reviews to predict the helpfulness, especially for unreliable and cold-start reviews. Furthermore, we present a detailed analysis to interpret the model behavior and provide reasoning behind model predictions. For future work, we will look into the problem of personalized review helpfulness prediction to model the demographics and cultural differences of the reviewers.
**Limitations**

Despite the effectiveness of incorporating the reviewer’s history and temporal information of the reviews in helpfulness prediction, our current studies still have several limitations, which can pave the path for future research.

For simplicity, like existing works, we assume that all the users rate reviews unanimously. However, the diversity of demographics, age, and cultural background also affect how users give, receive, and understand the sentiments expressed in reviews. Users may focus on different review aspects based on their preferences (i.e., “5 stars, party every night” vs “5 stars, always quiet and peaceful”). It would be interesting to see how to incorporate personal preferences for the helpfulness prediction task.

Another limitation of our work is that we only worked with reviews written in English. As a result, we filter out the reviews written in other languages and notice code-switched reviews when the reviewers alternate between two or more languages in a single review. We aim to extend this work to support more languages.

**Ethics Statement**

In our data scraping process, we took into account ethical considerations. We obtained data at an appropriate pace, avoiding any potential DDoS attacks. Additionally, we eliminated any Personal Identifying Information, such as names, telephone numbers, and email addresses, from the data set.

**Acknowledgements**

We thank all the anonymous reviewers for their valuable feedback and constructive suggestions for improving this work. This research is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and by a grant from Huawei. Mir Tafseer Nayeem is also supported by a Huawei Doctoral Scholarship.

**References**


A Dataset Construction

Publicly available datasets which are mostly used for this task are Amazon\(^8\) (He and McAuley, 2016; McAuley et al., 2015) and Yelp\(^9\). In Yelp dataset, the user votes are distributed among three categories such as "Useful", "Funny" or "Cool", where "Useful" voting feature was introduced much later than the other two categories. Therefore, many good reviews already in the dataset may not have been marked useful. On the other hand, the Amazon dataset does not contain the reviewers’ reviewing history and helpfulness votes to evaluate our hypothesis studied in this paper. Moreover, for Amazon, the samples come from various categories such as Books, Electronics, Clothing, Beauty, Shoes and Jewelry, Grocery, Pet Supplies, etc – the total helpfulness votes for the reviewers are coming from different categories and it’s not explicit in the fields from Amazon website. Therefore, it’s hard to devise expertise because of domain diversity.

We build our dataset by scraping reviews from TripAdvisor\(^10\), a travel site that offers online hotel and restaurant reservations and a platform for sharing the travel experiences of users. We take reviews from January 1st, 2015 until January 1st, 2020, and extract only those written in English. For each review, we extract the review text, the total helpfulness votes and the posting time, and for each reviewer, we extract the number of reviews contributed and the cumulative helpfulness votes. The attributes we extracted are summarized as follows:

- Reviews
  - Review Text
  - Total Review Helpful Votes
  - Review Posting Time
- Reviewers
  - Total Number of Reviews Contributed
  - Cumulative Helpful Votes

B Baseline Systems

We compare our system performance with the following baselines.

- ARH (Kim et al., 2006) & UGR + BGR (Xiong and Litman, 2011) use machine learning-based methods with hand-crafted features such as structural, lexical, syntactic, emotional, semantic, and meta-data from the review text to address this task. These features are fed into conventional classifiers such as SVM, Random Forest, and gradient boosting to identify helpful reviews.
- TextCNN (Chen et al., 2018) employs a text-based CNN model (Kim, 2014) to automatically capture the character-level, word-level, and topic-level features for helpfulness prediction.
- MTNL (Fan et al., 2018) utilizes end-to-end multi-task neural learning (MTNL) architecture for classifying helpful reviews. They take the help of an auxiliary task, such as rating regression, to boost the performance of the original task, which is review helpfulness identification.
- BERTHelp (Xu et al., 2020) develop their helpfulness prediction model using pre-trained BERT (Devlin et al., 2019). They design a regression model using BERT-based features extracted from review texts, star rating, and product type information from Amazon product review dataset (He and McAuley, 2016).

C Preprocessing & Hyperparameters

Preprocessing We didn’t perform any explicit preprocessing of the review text. Instead, we use BertTokenizer to avoid the out-of-vocabulary (OOV) problem, which uses WordPiece (Wu et al.,
2016) for tokenizing the sentences into words or subwords. In addition, we add special tokens to the start (e.g., [CLS]) and end of each review text (e.g., [SEP]) and truncate all sentences to a single constant length (e.g., 512).

**Hyperparameters** We use Adam optimizer (Kingma and Ba, 2015) with a learning rate of $3 \times e^{-5}$ and a batch size of 32. We use BERTBASE (Wolf et al., 2020) pre-trained model with a fixed vocabulary. We run the training for 5 epochs and check the improvement of validation (dev set) loss to save the latest best model during training.