International Journal of Networking and Computing – www.ijnc.org, ISSN 2185-2847
Volume 14, Number 1, pages 93–106, January 2024

Overall Rating Prediction from Review Texts using Category-oriented Japanese Sentiment Polarity Dictionary

Zaku Kusunoki, Sayaka Kamei and Yasuhiko Morimoto
Graduate School of Advanced Science and Engineering, Hiroshima University,
Kagamiyama 1-7-1, Higashi-Hiroshima 739-8521, Japan
{m224597, s10kamei, morimo}@hiroshima-u.ac.jp

Received: February 11, 2023
Revised: May 4, 2023
Revised: June 23, 2023
Accepted: July 10, 2023
Communicated by Hiroyuki Sato

Abstract

Hotel booking sites provide evaluations, including textual reviews and numerical ratings by hotel guests. However, some evaluations do not include numerical ratings, and there are some evaluations in which textual reviews and numerical ratings are inconsistent (i.e., a positive review text is posted along with a low rating, or vice versa). Such evaluations may need to be clarified for site users. To resolve such problems, we propose three highly accurate methods to predict an overall numerical rating from a textual review. Our new proposal is to use *Category-oriented Sentiment Polarity Dictionaries* (CSPD), which are automatically compiled for each category using a Rakuten Travel review database. The CSPD gives the *sentiment polarity value* (i.e., the positivity/negativity value) for each sentiment word for each category. Our proposed methods first predict category ratings from the BERT vector for the review and the CSPD. After that, based on the predicted category ratings and the BERT vector, our methods predict the overall rating. We conducted evaluation experiments using the Rakuten Travel review dataset for 2014-2019. Our experimental results show that our methods achieve higher accuracy than using only BERT vectors and successfully detect inconsistent evaluations.

Keywords: Rating Prediction, Natural language processing, Sentiment analysis, BERT

1 Introduction

Many hotel booking sites (e.g., Rakuten Travel [4], TripAdvisor [5], etc.) have both textual *reviews* and numerical *ratings*. In the textual reviews, reviewers describe their impressions of the hotels where they stayed, and in the numerical ratings, they express their satisfaction with the hotels. Because numerical ratings are more intuitive than verbal reviews, viewers of such sites often rely solely on rating values to get information about hotels. On the other hand, because each reviewer has their own way of assigning ratings, some evaluations give very different impressions to site viewers, despite being for the same hotel or by the same reviewer. Additionally, there are some evaluations in which textual reviews and numerical ratings are inconsistent. In other words, in some evaluations, positive review texts are posted along with low ratings, or vice versa. Such evaluations can be confusing to site viewers.

[0]This paper is an extended version of our one published in [13].

In this paper, we consider the prediction of ratings from textual reviews. By predicting the rating values from the reviews, we can alert a reviewer based on the predicted rating values from his review, if his rating values are extremely far apart from the contents of his review. Such alerts may cause reviewers to reconsider their rating values and review contents, and consequently can reduce the number of incorrectly chosen rating values and make textual reviews more detailed. In addition, because the reliability of these inconsistent reviews may be doubtful, the rating prediction can be used to detect spam reviews. Thus, such tasks are helpful not only for site users but also for hotel owners, enabling them to know the opinions of their guests.

We focus on predicting overall rating values from Japanese reviews on Rakuten Travel's hotel booking site [4]. On the Rakuten Travel site, in addition to giving an *overall rating*, each hotel guest can also provide a rating for each of the following six viewpoints: "*Service*", "*Location*", "*Rooms*", "*Facilities and amenities*", "*Baths*", and "*Meals*." We call these viewpoints *categories*. It is thought that each reviewer determines an overall rating based on the categories that they consider essential. In addition, just as reviewers write in their review text their impressions not only regarding these six categories but also regarding various other perspectives, the overall ratings can indicate their satisfaction with all viewpoints.

In our proposed methods, we use BERT (Bidirectional Encoder Representations from Transformers) [9] to characterize each word of a textual review as a numerical vector. However, it is difficult to pick up each word's positive/negative sentiment in the word vectorization, although such sentiments seem essential for the prediction of ratings. The degree to which each word in a sentence is negative or positive is called a *sentiment polarity value*, and words with sentiment polarity values are called *sentiment words*. To deal with such polarities, some *Sentiment Polarity Dictionaries* (SPDs), which contain pairs of sentiment words and sentiment polarity values, are used, e.g., [22], [2], and [20]. By comparing the words of a sentence to the SPD, we can indicate how much the sentence represents praise or criticism. However, these SPDs are not specialized for review texts. Because the sentiment polarities of the words vary according to the categories described in reviews [18], the values of the standard SPDs are not appropriate for reviews.

We believe that the accuracy of predicting overall ratings can be improved by using an SPD specific to each category described in hotel reviews. Thus, we use a *Category-oriented SPD* (CSPD) proposed by Shibata et al. [17, 18]. They calculated the sentiment polarity value of each word for each category based on the appearance rate of the word in the hotel reviews for each rating value of each category.

In this paper, we propose methods to predict overall ratings using the review vectors obtained by BERT and the category sentiment polarity values based on the CSPD for each of the six categories. While the overall ratings can indicate the reviewer's satisfaction not only in each of the six categories but also regarding various other perspectives, the satisfaction concerning the six categories should have no small effect. The results of our experiments show that our methods achieve higher accuracy than using only BERT vectors. Notably, we found that it is effective to predict category ratings using only the sentences mentioned for each category and to use the predicted values of the six category ratings to predict the overall ratings. In addition, according to the results of the questionnaire survey on reviews with significant differences between reviewers' ratings and predicted ratings, our proposed method successfully detected inconsistent reviews.

2 Related Work

2.1 Prediction of Ratings from Hotel Reviews

Various methods exist for the prediction of overall ratings from review texts by considering category ratings, e.g., [11], [23], and [24]. Fujitani et al. [11] used Bag-of-Words as the word vectorization method. They used multi-instance multi-label learning for relation extraction [19], and logistic regression [8] to classify Japanese review texts. They divided each review into sentences, predicted category ratings and an overall rating for each sentence, and then predicted the overall rating value using those predicted sentence ratings. In their method, the user information (such as the user ID, the purpose of the trip, and the companion) and the number of occurrences of each part of speech

Table 1: Examples of CSPD (In fact, each word is in Japanese.).

	Service	Location	Rooms	Facilities	Baths	Meals
messy	−4.7	−5.1	−7.8	−7.5	−7.2	−4.6
light	−2.2	1.3	−0.5	0.6	−0.6	0.05
delicious	3.4	no polarity	3.4	3.2	3.7	4.1
clean	2.6	2.1	4.6	3.6	3.2	3.1
⋮	⋮	⋮	⋮	⋮	⋮	⋮

Table 2: Words breakdown of CSPD.

	Service	Location	Rooms	Facilities	Baths	Meals
#words	3,849	3,173	3,866	3,574	3,013	3,242

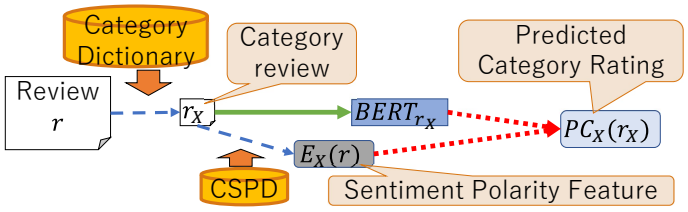

Figure 1: Shibata et al.'s method to predict category ratings.

in the review sentences were added as features contributing to the prediction of the rating values of the segmented sentences and the overall rating value. Toyama et al. [23] also considered Japanese review texts. In their method, each review and sentence was vectorized by PV-DM [14], which can generate a distributed representation of sentences and documents considering the order of words, and input this to a neural network classifier to predict category ratings and an overall rating. Chuhan et al. [24] predicted the category and overall ratings using CNN as the word encoder and the classifier for Chinese review texts.

In predicting the rating values from the content of the reviews, it is essential to consider the positive and negative sentiments expressed in the reviews. However, the word vectorization methods used by the methods described above cannot capture the sentiments expressed in the words of the reviews.

2.2 Category-oriented Sentiment Polarity Dictionaries

To predict the rating for each category (not the overall rating) of hotel reviews on Rakuten Travel, Shibata et al. [17, 18] created a category-oriented sentiment polarity dictionary (CSPD) in which sentiment words and their sentiment polarity values were registered for each category based on the appearance rate of the words in each category rating value. Unlike existing SPDs, each sentiment word has a real number as a sentiment polarity value, which is unique to each category in the CSPD. By means of this property, the CSPD corresponds to the different connotations of each word in the reviews, based on the different categories. For example, in the CSPD, while the sentiment polarity value of the word "spacious" is about 2 or 3 in the "Service", "Location", and "Meals" categories, it is about 6 in the "Rooms", "Facilities and amenities", and "Baths" categories. In the CSPD, the average sentiment polarity values of all words in each category are set closer to 0, their variance per category is set to 1, and their sentiment polarity values are from −7.81 to 6.38. Table 1 shows examples of CSPD (In reality, every word in the CSPD is in Japanese.), and Table 2 shows the word breakdown of the CSPD.

To predict a category rating value, Shibata et al. extracted sentences r_X (called a *category review*) for each category X from each review r, used their CSPD to extract a *Sentiment Polarity*

Feature (SPF) $E_X(r)$ from r_X, and added $E_X(r)$ to the sentence vector of r_X as a feature. They used BERT [9] as a word vectorization method for review sentence vectors and a Multinomial Logistic Regression (MLR) model [6][1] as a classifier. Figure 1 shows the flow of their method to predict category ratings. In this figure, dashed blue (resp. thick green) arrows represent applying dictionaries (resp. applying BERT), and dotted red arrows represent the inputs and outputs of the classifier to predict ratings. Shibata et al.'s method made predictions with higher accuracy than using only BERT vectors of r_X. In addition, the prediction accuracy was higher than using BERT vectors and existing SPDs, indicating that the CSPD is effective in the prediction of category ratings. However, they did not consider the overall rating values.

In this work, using their CSPD, we consider categories and sentiments represented in each review to predict its overall rating. In other words, we propose ways to use the CSPD to predict the overall rating.

3 Proposed Methods

In this study, we use the Rakuten Travel dataset [12] as a Japanese review dataset and discuss the prediction of overall ratings from the reviews. Note that example reviews in English shown in this section are actually in Japanese.

3.1 Overview of Proposed Methods

To predict an overall rating from each review r, we generate a vector of r and input it into the MLR model as a classifier, i.e., r is classified into one of the integer rating values from 1 to 5. The reason is that we believe that discrete numbers from 1 to 5 are more effective than real numbers as output ratings when considering the alert system introduced in Section 1. We propose three types of vectors of r as the inputs to the classifier, with a vector of each type using BERT, a pre-trained model in Japanese [3]. For each r, each word is transformed into its surface form and is vectorized by BERT with $1,024$ dimensions. After that, all vectors of the words in r are averaged. We call the average vector of r "$BERT_r$." Additionally, we derive the category review r_X from r by extracting sentences about each category X (see Section 3.2). Each proposed method combines $BERT_r$ and one of the following 6-dimensional vectors to predict the overall rating attached to r, i.e., we use one of the following vectors as an additional feature in each of our methods.

1. *Sentiment polarity vector of r: SPV_r.*
   SPV_r is a 6-dimensional vector, where each element is the SPF in a category. Based on the CSPD of category X, we calculate the SPF $E_X(r)$ of r in X by averaging the sentiment polarity values of sentiment words in r_X (see Section 3.3).

2. *Predicted category rating vector by r: PCV_r.*
   For category X, the category rating of r is predicted from a $1,025$-dimensional vector consisting of $BERT_r$ and $E_X(r)$. We obtain the predicted category rating for each of the six categories and combine them into a 6-dimensional vector PCV_r (see Section 3.4).

3. *Predicted category rating vector by r_X: PCV_{r_X}.*
   To predict a category rating for r in X, we use $BERT_{r_X}$, a vector of the category review r_X using BERT. Like $BERT_r$, for each r_X, each word is vectorized by BERT with $1,024$ dimensions, and all vectors of the words in r_X are averaged. Using the method of Shibata et al. [18], we compute the predicted category rating of X from a vector of $1,025$ dimensions, consisting of $BERT_{r_X}$ and $E_X(r)$. We obtain the predicted category ratings for each of the six categories and combine them into a vector PCV_{r_X} in 6-dimensions (see Section 3.4).

[1]The MLR is an extension of the original logistic regression [8] to classify the multiple classes. Because the MLR finds the predicted probability of each class using the softmax function, each review is assigned to the class with the highest probability.

Table 3: Example of the category dictionary (In fact, each word is in Japanese).

Service	Location	Rooms	Facilities	Baths	Meals
courtesy	station	TV	stairs	hot spring	dessert
cancel	access	carpet	elevator	sauna	buffet
price	sightseeing	bed	towel	open-air bath	food
⋮	⋮	⋮	⋮	⋮	⋮

Table 4: Words breakdown of the category dictionary.

	Service	Location	Rooms	Facilities	Baths	Meals	Total
#words	74	50	97	89	30	56	330

When we combine these vectors with $BERT_r$, we call the obtained $1,030$-dimensional vectors $BERT_r$+SPV_r, $BERT_r$+PCV_r, and $BERT_r$+PCV_{r_X}, respectively. In the following section, we explain the method of obtaining these 6-dimensional vectors in detail.

3.2 Generating Category Reviews

First, we split each review r in the dataset into sentences based on a punctuation mark indicating the end of the sentence, such as '.', '!', '?', etc.. Using Janome [1], a Japanese morphological analysis engine, each review sentence is divided into words, and all the words are transformed into their base forms.

The segmented review sentences are then classified into six categories using a category dictionary created by Shibata et al. [18][2], that compiles words related to each category. Examples of elements of the category dictionary are shown in Table 3, and a breakdown of words in the category dictionary is shown in Table 4. If a sentence contains a word existing in the category dictionary, then the category label is assigned to the sentence. A sentence containing words from several categories is assigned to these multiple categories. From each review r, we extract sentences with the label of category X and combine them into a set of sentences called a category review r_X.

For example, consider the review "The open-air bath was wonderful. There was a sauna. The food was delicious." Because the words "open-air bath" and "sauna" are in the "Baths" category, and "food" is in the "Meals" category in the category dictionary, the first two sentences become a category review of the "Baths" category, and the last sentence becomes a category review of the "Meals" category.

3.3 Sentiment Polarity Feature Extraction

SPFs are extracted from each category review r_X using the CSPD of category X. We assign a sentiment polarity value to all sentiment words in r_X. In CSPD of X, let E_{X_w} be the sentiment polarity value of the sentiment word w. Then, if the next word after w in r_X is the negative word "*nai* (not)," we regard $E_{X_w} \times (-1)$ as the sentiment polarity value of w in r_X. Then, the average sentiment polarity value in r_X is calculated as the SPF $E_X(r)$ of r. When calculating $E_X(r)$, it is treated as 0 if there is no sentiment word of X in r_X or there is no r_X in r.

For example, consider the category review, "The miso soup was lukewarm and not tasty." for the "Meals" category. In the "Meals" category of CSPD, the word "lukewarm" has -3.94, and "tasty" has 3.67. Then, the SPF of this category review is $(-3.94 + 3.67 \times (-1))/2 = -3.805$.

Combining the computed sentiment features $E_X(r)$ for each of the six categories, we create a 6-dimensional vector SPV_r. Figure 2 shows the flow of the proposed method $BERT_r$+SPV_r. In this figure, the meanings of the arrows are the same as in Figure 1.

[2]While the original category dictionary was created by Takuma et al. [21], Shibata et al. manually derived only words such that, if each word is contained in a review, we can recognize the category described in the review. They also added some words. Using their category dictionary, Shibata et al. showed that 90.83% of category reviews were with correct labels.

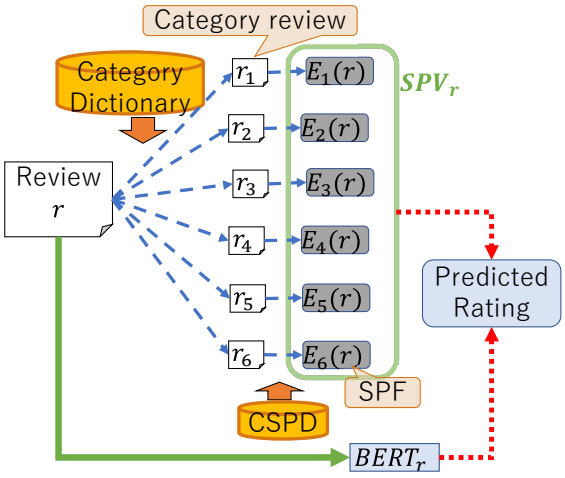

Figure 2: Proposed method $BERT_r$+SPV_r.

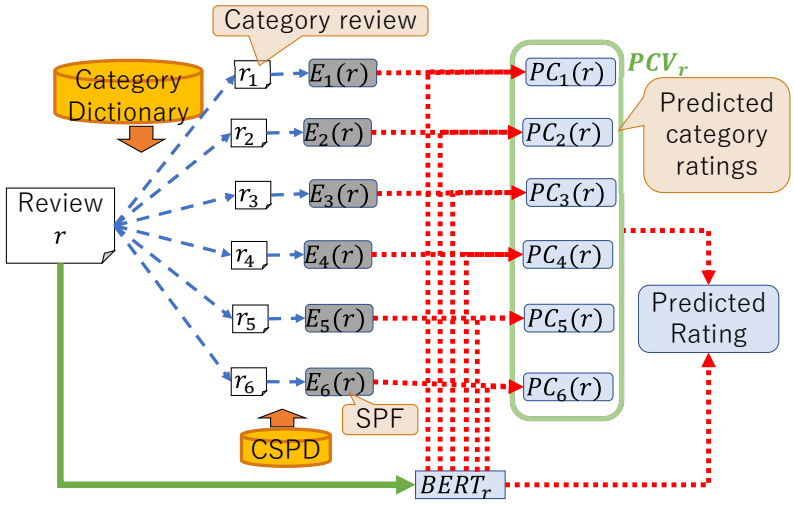

Figure 3: Proposed method $BERT_r$+PCV_r.

3.4 Generating Predicted Category Rating Vectors

Next, we calculate the predicted category rating for each category X. The predicted category rating of X by r (resp. r_X) is calculated using a 1025-dimensional vector such that $E_X(r)$ is added to $BERT_r$ (resp. $BERT_{r_X}$) as an additional feature. Then, the obtained vector is input into the MLR model to predict the category ratings. We denote $PC_X(r)$ (resp. $PC_X(r_X)$) as the predicted category rating for r (resp. r_X) in X.

Note that the method of calculating $PC_X(r_X)$ for each category X is the same as the method proposed by Shibata et al. [18]. However, if r does not contain any description about X (i.e., r_X does not exist for r), we cannot obtain $PC_X(r_X)$. In such cases, we average the category ratings of X by the reviewers for the hotel and substitute that value as $PC_X(r_X)$.

By combining the six predicted category ratings, we create a 6-dimensional vector PCV_r (resp. PCV_{r_X}). Then, the obtained vector and $BERT_r$ are input into the MLR model to predict the overall ratings. In Figure 3 (resp. Figure 4), we show the flow of the proposed method $BERT_r$+PCV_r (resp. $BERT_r$+PCV_{r_X}). In these figures, the meanings of the arrows are the same as in Figure 1.

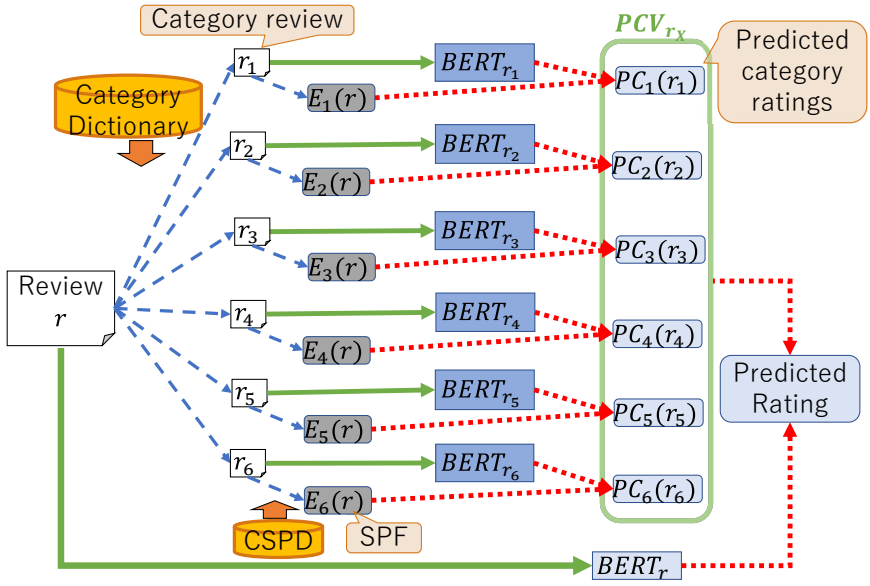

Figure 4: Proposed method $BERT_r$+PCV_{r_X}.

4 Evaluation Experiments

In our experiments, we used the Rakuten Travel dataset [12] of 2014-2019. The reviews in this dataset are written mainly in Japanese, but the dataset also includes a small number of reviews written in English.

4.1 Experiment 1: Using Random Sampling Data

First, we used random sampling data as usual.

4.1.1 Dataset and Implementation

From the dataset of 2014-2016, we derived only reviews with an overall rating and six category ratings for the hotels which received at least 100 reviews in 2014-2016, and we call this derived dataset "the old dataset". There are 313,891 reviews in the old dataset. Table 5 shows the number of reviews for each overall rating in the old dataset. These reviews were split into 1,188,775 sentences. While we assigned category labels for each sentence, 423,879 sentences were not assigned any category labels. In other words, about 35.7% of split sentences were not able to be in category reviews. Table 6 shows the number of category reviews in the old dataset.

From the old dataset, we first randomly derived 70,000 category reviews for each category as training data for the prediction model of category ratings in PCV_r and PCV_{r_X}. Then, for PCV_r, we used the original review r from which the selected category review r_X was derived. Using the training data for each category, we made a classifier model to predict category ratings. After that, we extracted 70,000 reviews randomly from the old dataset as training data for the model to predict overall ratings. Using the six prediction models for category ratings, we made a classifier model to predict an overall rating for each method. We repeated these processes (i.e., from sampling category reviews to training a model to predict an overall rating) five times.

To predict the overall ratings and the category ratings, we used the MLR of scikit-learn [16] in the default settings[3]. Then, we let the rating by the reviewer be the correct answer and the predicted value by the classifier be the predicted rating.

[3]By default, the improved version of the memory-constrained BFGS method [7,15] is used for optimization.

Table 5: The number of reviews for each rating in the old dataset.

Overall rating	1	2	3	4	5	Total
#reviews	4,254	9,125	32,799	136,056	131,657	313,891
#reviews written about 6 categories	262	622	1,635	6,332	7,220	16,071

Table 6: The number of category reviews in the old dataset.

Category	Service	Location	Rooms	Facilities	Baths	Meals
#category reviews	173,425	127,837	172,906	114,254	113,292	193,718

Table 7: The number of reviews for each rating in the new dataset.

Overall rating	1	2	3	4	5	Total
#reviews	4,681	10,292	34,706	144,182	148,396	342,257
#reviews written about 6 categories	302	651	1,494	5,633	6,955	15,035

Table 8: The number of reviews for each rating in the test data.

Overall rating	1	2	3	4	5	Total
#reviews	149	310	1,030	4,251	4,260	10,000

Table 9: The number of category reviews in the test data.

Category	Service	Location	Rooms	Facilities	Baths	Meals
#category reviews	5,104	3,812	5,287	3,459	3,672	6,185

Table 10: The number of category reviews per review in the test data.

#category	0	1	2	3	4	5	6
#reviews	758	1,765	2,087	2,043	1,742	1,147	458

Table 5 also shows a breakdown of reviews, including descriptions of all six categories, in the old dataset. Note that if a review r did not contain a description of category X, then we substituted the average of the category ratings of X by the reviewers for the hotel rated by r during 2014-2016 for $PC_X(r_X)$. When considering the actual application, it is necessary to use such substitutions because only about 5% of all reviews mention all six categories.

From the dataset of 2017-2019, we also derived only reviews with an overall rating and six category ratings for the hotels which received at least 100 reviews in 2014-2016. We call this derived dataset "the new dataset". There are $342,257$ reviews in the new dataset, and Table 7 shows their breakdown. For the test data, we randomly extracted $10,000$ reviews from the new dataset. Table 8 shows the number of reviews for each overall rating, and Table 9 shows the number of category reviews, in the extracted test data. Table 10 shows how many categories are described in each review in the test data. It shows that only 458 reviews mention all six categories, and the average number of categories mentioned per review is 2.75, even though the reviews have all six category rating values.

4.1.2 Evaluation

For comparison, in addition to our three proposed methods, we also created models to predict the overall rating using only $BERT_r$, SPV_r, PCV_r, and PCV_{r_X}, respectively, as inputs to the classifier. The prediction results made by each of the four methods were evaluated using the following five indices. Let n be the number of predictions, y_i $(1 \leq y_i \leq 5)$ be the value of the correct rating by the reviewer, $\hat{y}_i$ $(1 \leq \hat{y}_i \leq 5)$ be the predicted rating, where $1 \leq i \leq n$. Let R_y be the set of reviews with correct rating y $(1 \leq y \leq 5)$, and $\hat{R}_y$ be the set of reviews with predicted rating $\hat{y}$ $(1 \leq \hat{y} \leq 5)$. Let C be the set of reviews such that $y_i = \hat{y}_i$ holds.

- Accuracy: The ratio of correctly predicted reviews, i.e., $|C|/n$.
- Precision: The ratio of reviews that resulted in C of reviews in $\hat{R}_y$, i.e., $|C \cap \hat{R}_y|/|\hat{R}_y|$.
- Recall: The ratio of reviews that resulted in C of reviews in R_y, i.e., $|C \cap R_y|/|R_y|$.
- F1: The harmonic mean of precision and recall.
- RMSE: The root mean square error between the correct and predicted ratings, i.e.,

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}.$$

The precision, recall, and F1 were calculated for each rating value, and a weighted average value was used since Table 8 showed that the number of reviews was biased for each rating. In the following sub-section, we compare results using the average value of each evaluation index from five classifier models for each of the four methods.

4.1.3 Results of Evaluation

Table 11 shows the results. In the table, the best performance of each evaluation index is emphasized in bold. We performed t-tests on the results; $**$ (resp. $*$) denotes $p \leq 0.01$ (resp. $p \leq 0.05$) when compared with $BERT_r$. Table 11 shows that $BERT_r$ performed better than using only SPV_r (resp. PCV_r, PCV_{rx}). On the other hand, $BERT_r$+SPV_r and $BERT_r$+PCV_{rx} performed better than using only $BERT_r$, while there was no significant difference between $BERT_r$+SPV_r and $BERT_r$. Significantly, $BERT_r$+PCV_{rx} outperformed others.

In addition, Table 12 (resp. Table 13) shows the confusion matrix of the predictions by $BERT_r$ (resp. $BERT_r$+PCV_{rx}). In these confusion matrices, each value represents the average percentage of each predicted rating for each correct rating among five classifier models. The highest ratio in each correct (accurate) rating is emphasized in bold. Note that each diagonal value (i.e., where the correct and predicted ratings are the same) is the recall value for the rating. The one with higher recall between the two methods is indicated by †. Table 14 shows the precision and F1 values for each rating. We also performed t-tests on the recall, precision, and F1 values, respectively; in Tables 13 and 14, $**$ (resp. $*$) denotes $p \leq 0.01$ (resp. $p \leq 0.05$) when compared with $BERT_r$. Although one of the reasons for the poor accuracy of the rating values 1 and 2 for both methods is that the ratings in the training data are highly biased, these confusion matrices show that $BERT_r$+PCV_{rx} is better concerning each rating value larger than or equal to 3. Table 14 shows that, except for rating value 2, the precision of $BERT_r$+PCV_{rx} is better.

4.2 Experiment 2: Mitigated Rating Bias

There is a significant rating bias in the training data of the previous experiment in Section 4.1. Specifically, the number of reviews with low ratings is very small. None of the methods can predict these low ratings.

Thus, in this section, to mitigate rating bias, we used every review with ratings of 1 or 2 in the old dataset (Table 5) as a part of the training data for each classifier model of overall ratings. From reviews with ratings of 4 and 5 in the old dataset, we randomly extracted $20,000$ reviews for each of the ratings. We randomly selected the remaining $16,621$ reviews from the reviews with ratings of 3 in the old dataset. The breakdown of the training data is shown in Table 15. Note that, we use the same classifier models to predict category ratings for each category in Section 4.1. In other words, we repeated the following processes five times: selecting one of the sets of the category rating prediction models (each set is used at most once), generating training data for the overall rating prediction model, and training the overall rating prediction model.

We used the same test data as in Section 4.1.1 (Tables 8-10), and evaluated the prediction results using the same five indices as in Section 4.1.2. For the comparison, in addition to our three proposed methods, we also created models using only $BERT_r$.

Table 11: Results of Experiment 1.

	Accuracy	Precision	Recall	F1	RMSE
$BERT_r$	0.5837	0.5754	0.5837	0.5760	0.7436
SPV_r	0.4932**	0.4717**	0.4932**	0.4593**	0.8843**
PCV_r	0.5790**	0.5660**	0.5790**	0.5641**	0.7568**
PCV_{r_X}	0.5577**	0.5415**	0.5577**	0.5349**	0.7945**
$BERT_r$+SPV_r	0.5840	0.5757	0.5840	0.5766	**0.7419**
$BERT_r$+PCV_r	0.5830	0.5751	0.5830	0.5761	0.7436
$BERT_r$+PCV_{r_X}	**0.5888****	**0.5796***	**0.5888****	**0.5809****	0.7420

Table 12: Confusion matrix of $BERT_r$ (Experiment 1).

		Predicted Rating				
		1	2	3	4	5
Correct Rating	1	0.3678 †	**0.3839**	0.1154	0.1087	0.0242
	2	0.1110	0.2710 †	0.2477	**0.3394**	0.0310
	3	0.0190	0.0685	0.2243	**0.6140**	0.0742
	4	0.0034	0.0087	0.0503	**0.6238**	0.3138
	5	0.0007	0.0013	0.0108	0.3263	**0.6609**

Table 13: Confusion matrix of $BERT_r$+PCV_{r_X} (Experiment 1).

		Predicted Rating				
		1	2	3	4	5
Correct Rating	1	0.3503	**0.3544**	0.1463	0.1221	0.0268
	2	0.1065	0.2426*	0.2748	**0.3361**	0.0400
	3	0.0175	0.0658	0.2351* †	**0.6066**	0.0750
	4	0.0033	0.0083	0.0501	**0.6239** †	0.3143
	5	0.0005	0.0014	0.0099	0.3154	**0.6728**** †

Table 14: Precision and F1 value for each ratings (Experiment 1).

	Correct Rating	1	2	3	4	5
Precision	$BERT_r$	0.4358	**0.3301**	0.3954	0.5529	0.6642
	$BERT_r$+PCV_{r_X}	**0.4382**	0.3166	**0.4012**	**0.5592****	**0.6672**
F1	$BERT_r$	**0.3975**	**0.2972**	0.2861	0.5862	0.6626
	$BERT_r$+PCV_{r_X}	0.3882	0.2743	**0.2963***	**0.5898**	**0.6700****

4.2.1 Results of Evaluation

Table 16 shows the results of Experiment 2. In the table, the best performance of each evaluation index is emphasized in bold, and ** (resp. *) denotes the results of t-tests, $p \leq 0.01$ (resp. $p \leq 0.05$) when compared with $BERT_r$. Table 16 shows that our three proposed methods performed better than using only $BERT_r$. Compared to Experiment 1, because $BERT_r$+PCV_r became better than $BERT_r$ in Experiment 2, $BERT_r$+PCV_r may be said to be more influenced by rating bias. Even in this case, $BERT_r$+PCV_{r_X} outperformed others.

In addition, Table 17 (resp. Table 18) shows the confusion matrix of the predictions by $BERT_r$ (resp. $BERT_r$+PCV_{r_X}). Table 19 shows the precision and F1 values for each rating. In Tables 18 and 19, ** (resp. *) also denotes the results of t-tests, $p \leq 0.01$ (resp. $p \leq 0.05$) when compared with $BERT_r$. The confusion matrices show that $BERT_r$ had better recall concerning each rating value of 1-3. However, the precision of $BERT_r$+PCV_{r_X} for each rating was higher than that of $BERT_r$. Note that, compared with Experiment 1, although every evaluation index in Table 11 was better than in Table 16, the F1 values of each rating other than 4 based on both methods in Table 19 became better than in Table 14.

Table 15: The number of reviews for each rating in the training data for Experiment 2.

Overall rating	1	2	3	4	5	Total
#reviews	4,254	9,125	16,621	20,000	20,000	70,000

4.2.2 Reviews with Large Differences between Correct Ratings and Predicted Ratings

Next, we observe reviews in which the polarities (positive/negative) between the reviewer's rating and the predicted rating differed. In other words, we consider the reviews such that the difference between the reviewer's rating and the prediction by at least one of the five models of each of $BERT_r$+PCV_{rx} and $BERT_r$ was greater than 2, but these ratings were not 3. Table 20 shows the breakdown of such Japanese reviews. For $BERT_r$+PCV_{rx} (resp. $BERT_r$), there are 284 (resp. 273) distinct reviews. Note that, as predicted ratings for some reviews differed among five models for each method, the total number of "#reviews" and the total number of "#distinct reviews" are not equivalent in this table. In the parentheses of "#distinct reviews", the numbers indicate the numbers of distinct reviews detected only by the method, i.e., these reviews are not included in the set of distinct reviews for the other method. We call a set of reviews such that the reviewer's rating is negative (resp. positive) and the prediction is positive (resp. negative) NasP(Negative as Positive) (resp. PasN(Positive as Negative)). In the following experiment, we used all 63 reviews in NasP. For PasN, we used 94 reviews such that 34 reviews were detected only by one of the two methods, and 60 reviews were randomly extracted from reviews that were commonly detected by both methods.

For each review, three subjects were assigned to judge whether the reviewer's rating or the predicted rating was correct. They were asked to choose whether each review was positive, negative, or neutral, rather than give specific rating values. The instructions to the subjects were as follows: "Please read the following reviews for various hotels and choose whether each review is positive, negative, or neutral."

Table 21 shows the breakdown of responses from subjects by majority vote. However, if subjects' responses were entirely discrepant, the review was considered neutral. Note that, since the number of reviews detected by each method differed, we also show the ratio of each responses from the subjects for each type of reviews in this table, and the highest ratio for each type is emphasized in bold. Then, the values with $*$ represent that the ratio of incorrect predictions for the review contents. We calculated Fleiss' kappa value [10][4] for subjects' responses; it was 0.4937. This value means that the responses of the three subjects in each review are moderately in agreement.

First, we consider the reviews that were detected only by one of the two methods. Note that the numbers of such reviews are written in parentheses in Table 20. In NasP, the predictions by both methods were incorrect for the review content. In PasN, while 77.3% of the reviews detected by only $BERT_r$ gave negative/neutral impressions to the subjects, 58.3% of the reviews detected by only $BERT_r$+PCV_{rx} gave positive impressions to the subjects. However, it is difficult to compare these results because the number of these reviews is very small and varies between methods.

Next, we consider reviews that were commonly detected by both methods. Note that, we consider 32 (resp. 60) reviews in NasP (resp. PasN), and the "Intersection" part in Table 21 shows the results. Among the reviews detected by both methods in NasP, 43.8% of reviews gave incorrect impressions to the subjects (i.e., they had negative reviewer ratings, but their contents were not negative). In addition, in PasN, 91.67% of the reviews gave the subjects negative/neutral impressions. In other words, using only the intersection part, we can successfully detect inconsistent reviews given positive reviewer ratings with high precision.

[4]Fleiss' kappa value measures subjects' agreement, with a score of 1 indicating perfect agreement between subjects' responses and a score of 0 indicating that the responses were scattered.

Table 16: Results of Experiment 2.

	Accuracy	Precision	Recall	F1	RMSE
$BERT_r$	0.5382	0.5676	0.5381	0.5449	0.8170
$BERT_r$+SPV_r	0.5398*	0.5695*	0.5398*	0.5467*	0.8149
$BERT_r$+PCV_r	0.5402	0.5382	0.5402	0.5467	0.8113**
$BERT_r$+PCV_{r_X}	**0.5472****	**0.5720***	**0.5472****	**0.5531****	**0.8022****

Table 17: Confusion matrix of $BERT_r$ (Experiment 2).

		Predicted Rating				
		1	2	3	4	5
Correct Rating	1	0.4121 †	**0.5181**	0.0430	0.0134	0.0134
	2	0.1265	**0.4400** †	0.3703	0.0413	0.0219
	3	0.0216	0.1617	**0.5056** †	0.2406	0.0705
	4	0.0046	0.0305	0.2163	**0.4301**	0.3185
	5	0.0012	0.0056	0.0626	0.2651	**0.6654**

Table 18: Confusion matrix of $BERT_r$+PCV_{r_X} (Experiment 2).

		Predicted Rating				
		1	2	3	4	5
Correct Rating	1	0.4094	**0.5047**	0.0497	0.0148	0.0215
	2	0.1174	**0.4258***	0.3671	0.0561	0.0335
	3	0.0214	0.1571	**0.4948***	0.2555	0.0713
	4	0.0042	0.0272	0.2060	**0.4461****†	0.3164
	5	0.0011	0.0053	0.0516	0.2677	**0.6744***†

Table 19: Precision and F1 value for each rating (Experiment 2).

	Correct Rating	1	2	3	4	5
Precision	$BERT_r$	0.4166	0.2555	0.2848	0.5677	0.6639
	$BERT_r$+PCV_{r_X}	**0.4302***	**0.2603**	**0.2952***	**0.5713**	**0.6673***
F1	$BERT_r$	0.4143	**0.3233**	0.3644	0.4894	0.6647
	$BERT_r$+PCV_{r_X}	**0.4195**	0.3230	**0.3697**	**0.5010****	**0.6708***

5 Conclusion

This paper has proposed three methods for predicting overall ratings from hotel reviews. The results show that our methods effectively predict the overall ratings of hotel reviews when the rating bias is mitigated. Our proposed methods use BERT vectors and the predicted category ratings or sentiment polarity values for six categories. Our experiments found that it is effective to predict category ratings using only the sentences mentioned for each category, and to use the predicted category rating values to predict overall ratings.

However, the overall rating values indicate satisfaction concerning six categories and other various perspectives. Thus, in the future, we will consider how to extract these various perspectives from the reviews to improve the accuracy of the prediction.

Acknowledgment

In this paper, we used "Rakuten Dataset" (`https://rit.rakuten.com/data_release/`) provided by Rakuten Group, Inc. via IDR Dataset Service of National Institute of Informatics.

Table 20: Breakdown of the reviews with significantly incorrect prediction.

	Reviewer	Prediction	$BERT_r$+PCV_{r_X}		$BERT_r$	
			#reviews	#distinct reviews	#reviews	#distinct reviews
NasP	1	5	6	58 (26)	3	37 (5)
	1	4	4		6	
	2	5	15		11	
	2	4	40		27	
PasN	4	2	165	226 (12)	172	236 (22)
	4	1	24		24	
	5	2	36		36	
	5	1	5		6	
Total			295	284 (38)	285	273 (27)

( ) : The number of distinct reviews detected only by the method.

Table 21: Breakdown of responses from subjects.

		$BERT_r$+PCV_{r_X} only			$BERT_r$ only			Intersection		
		Negative	Neutral	Positive	Negative	Neutral	Positive	Negative	Neutral	Positive
NasP	#reviews	23	2	1	5	0	0	18	5	9
	Ratio(%)	**88.46***	7.69	3.85	**100***	0	0	**56.25***	15.63	28.13
PasN	#reviews	4	1	7	15	2	5	43	12	5
	Ratio(%)	33.33	8.33	**58.33***	**68.18**	9.09	22.73*	**71.67**	20.00	8.33*

References

[1] Janome(ja). `https://mocobeta.github.io/janome/`.

[2] Japanese Sentiment Polarity Dictionary. `https://www.nlp.ecei.tohoku.ac.jp/`. Inui-Suzuki Lab. in Tohoku University.

[3] Pretrained Japanese BERT models. `https://github.com/cl-tohoku/bert-japanese`. Tohoku NLP Group Github Repositoris.

[4] Rakuten Travel. `https://travel.rakuten.com/`.

[5] TripAdvisor. `https://www.tripadvisor.jp/`.

[6] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006.

[7] Richard H. Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific and Statistical Computing*, 16(5):1190–1208, 1995.

[8] David R. Cox. The regression analysis of binary sequences. *Journal of the Royal Statistical Society: Series B (Methodological)*, 20(2):215–232, 1958.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[10] Joseph L. Fleiss. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382, 1971.

[11] Yoshinori Fujitani, Makoto Miwa, and Yutaka Sasaki. Prediction of ratings for hotel reviews using hidden states. In *Proceedings of the 21th Annual meeting of the Association for Natural Language Processing*, pages 764–767, 2015. (in Japanese).

[12] Rakuten Group, Inc. Rakuten Dataset. Informatics Research Data Repository - National Institute of Informatics (dataset). `https://doi.org/10.32130/idr.2.0`, 2010.

[13] Zaku Kusunoki, Sayaka Kamei, and Yasuhiko Morimoto. Overall rating prediction from review texts using category-oriented japanese sentiment polarity dictionary. In *Proceedings of the Tenth International Symposium on Computing and Networking (CANDAR)*, pages 124–129, 2022.

[14] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31th International Conference on Machine Learning*, pages II–1188–II–1196, 2014.

[15] Dong C. Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45:503–528, 1989.

[16] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[17] Akito Shibata, Sayaka Kamei, and Koji Nakano. Category-oriented sentiment polarity dictionary for rating prediction of Japanese hotels. In *Proceedings of the 2020 Eighth International Symposium on Computing and Networking Workshops (CANDARW)*, pages 440–444, 2020.

[18] Akito Shibata, Sayaka Kamei, and Koji Nakano. Category-oriented Japanese sentiment polarity dictionary for rating prediction of hotels. *IPSJ TOD*, 14(3):16–29, 2021. (in Japanese).

[19] Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D. Manning. Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 455–465, 2012.

[20] Hiroya Takamura, Takashi Inui, and Manabu Okumura. Extracting semantic orientations of words using spin model. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 133–140, 2005.

[21] Koji Takuma, Junya Yamamoto, Sayaka Kamei, and Satoshi Fujita. A hotel recommendation system based on reviews: What do you attach importance to? In *Proceedings of the 4th International Symposium on Computing and Networking*, pages 710–712. IEEE, 2016.

[22] Mike Thelwall, Kevan Buckley, Georgios Paltoglou, Di Cai, and Arvid Kappas. Sentiment strength detection in short informal text. *Journal of the American Society for Information Science and Technology*, 61(12):2544–2558, 2010.

[23] Yota Toyama, Makoto Miwa, and Yutaka Sasaki. Rating prediction by considering relations among documents and sentences and among categories. In *Proceedings of the 22th Annual Meeting of the Association for Natural Language Processing*, pages 158–161, 2016. (in Japanese).

[24] Chuhan Wu, Fangzhao Wu, Junxin Liu, Yongfeng Huang, and Xing Xie. ARP: Aspect-aware neural review rating prediction. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2169–2172, 2019.