Online Appendix to
Mining Bilateral Reviews for Online Transaction Prediction: A Relational Topic Modeling Approach

Jiawei Chen
School of Information Management and Engineering, Shanghai University of Finance and Economics, chenjiawei@mail.shufe.edu.cn

Yinghui (Catherine) Yang
Graduate School of Management, University of California, Davis, yiyang@ucdavis.edu

Hongyan Liu
School of Economics and Management, Tsinghua University, liuhy@sem.tsinghua.edu.cn

Appendix A. Technical Details of Learning Algorithm

To learn our proposed model, we focus on computing the posterior distribution of the latent variables conditioned on the observed textual documents and links between them. Based on the notations used in Section 4.3 and 4.4, we can represent the posterior distribution as follows:

$$p(Z,\Theta,\Phi|W,Y,\alpha,\eta,\beta)=\frac{p(Z,\Theta,\Phi,W,Y|\alpha,\eta,\beta)}{p(W,Y|\alpha,\eta,\beta)},$$

where $Z=\{z^D,z^A,z^B\}$, $\Theta=\{\theta^D,\theta^A,\theta^B\}$, $\Phi=\{\phi^{D^*},\phi^{D\&A},\phi^{A^*},\phi^{A\&B},\phi^{B^*}\}$, $W=\{w^D,w^A,w^B\}$, $Y=\{y^1,y^2\}$, $\alpha=\{\alpha_D,\alpha_A,\alpha_B\}$, $\eta=\{\eta_{D^*},\eta_{D\&A},\eta_{A^*},\eta_{A\&B},\eta_{B^*}\}$, and $\beta=\{\beta_D,\beta_A,\beta_{C,H},\beta_1,\beta_B,\beta_{C,G},\beta_2\}$.

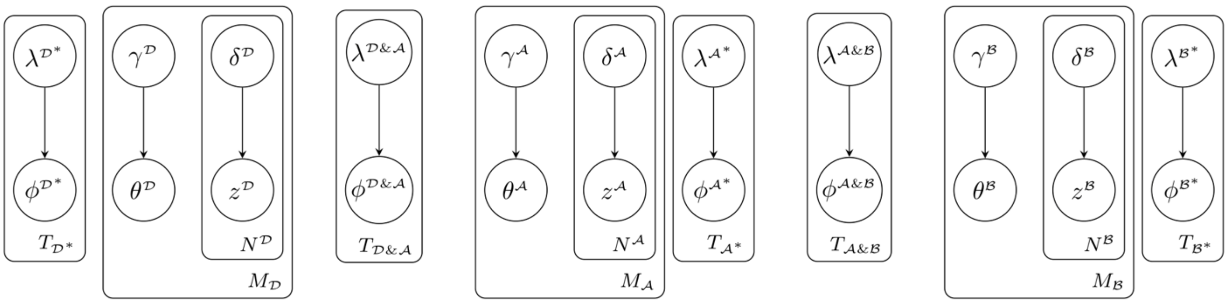
As with many hierarchical Bayesian models, exact posterior inference for our model is intractable to compute because of the coupling between Θ and Φ in the summation over latent topics (Blei et al. 2003). Thus, we approximate the inference using variational methods (Blei and Jordan 2006) with comparable accuracy and improved computational efficiency. We introduce a family of distributions over the latent variables indexed by free variational parameters. The parameters $\Gamma=\{\gamma^D,\gamma^A,\gamma^B\}$ are variational Dirichlet parameters, one for each document; $\Delta=\{\delta^D,\delta^A,\delta^B\}$ are variational multinomial parameters, one for each word in each document; and $\Lambda=\{\lambda^{D^*},\lambda^{D\&A},\lambda^{A^*},\lambda^{A\&B},\lambda^{B^*}\}$ are variational Dirichlet parameters, one for each latent topic. A summary of latent variables and corresponding variational parameters is displayed in Table A.1.

Table A.1. Summary of variational parameters.

Latent variables	Variational parameters	Description
$\Phi=\{\phi^{D^*},\phi^{D\&A},\phi^{A^*},\phi^{A\&B},\phi^{B^*}\}$	$\Lambda=\{\lambda^{D^*},\lambda^{D\&A},\lambda^{A^*},\lambda^{A\&B},\lambda^{B^*}\}$	Variational Dirichlet parameters, one for each latent topic.
$\Theta=\{\theta^D,\theta^A,\theta^B\}$	$\Gamma=\{\gamma^D,\gamma^A,\gamma^B\}$	Variational Dirichlet parameters, one for each type of documents.
$Z=\{z^D,z^A,z^B\}$	$\Delta=\{\delta^D,\delta^A,\delta^B\}$	Variational multinomial parameters, one for each word in each document.

With the introduction of variational parameters, all topic proportions, word distributions and topic assignments are considered independent as displayed in Figure A.1.

Figure A.1. Graphical representation of the variational distribution used to approximate the posterior.



Based on these parameters, we can obtain the following variational distribution:

$q(Z,\Theta,\Phi|\Gamma,\Delta,\Lambda) = q(\Phi|\Lambda)q(z^D,\theta^D|\delta^D,\gamma^D)q(z^A,\theta^A|\delta^A,\gamma^A)q(z^B,\theta^B|\delta^B,\gamma^B)$, where $q(\Phi|\Lambda) = \prod_{t=1}^{T_{D^*}} q\left(\phi_t^{D^*}|\lambda_t^{D^*}\right)\prod_{t=1}^{T_{D\&A}} q\left(\phi_t^{D\&A}|\lambda_t^{D\&A}\right)\prod_{t=1}^{T_{A^*}} q\left(\phi_t^{A^*}|\lambda_t^{A^*}\right)\prod_{t=1}^{T_{A\&B}} q\left(\phi_t^{A\&B}|\lambda_t^{A\&B}\right)\prod_{t=1}^{T_{B^*}} q\left(\phi_t^{B^*}|\lambda_t^{B^*}\right)$, $q(z^D,\theta^D|\delta^D,\gamma^D) = \prod_{m=1}^{M_D} q(\theta_m^D|\gamma_m^D)\prod_{n=1}^{N_m^D} q\left(z_{m,n}^D|\delta_{m,n}^D\right)$, $q(z^A,\theta^A|\delta^A,\gamma^A) = \prod_{m=1}^{M_A} q(\theta_m^A|\gamma_m^A)\prod_{n=1}^{N_m^A} q(z_{m,n}^A|\delta_{m,n}^A)$, and $q(z^B,\theta^B|\delta^B,\gamma^B) = \prod_{m=1}^{M_B} q(\theta_m^B|\gamma_m^B)\prod_{n=1}^{N_m^B} q(z_{m,n}^B|\delta_{m,n}^B)$.

These parameters are then optimized to minimize the Kullbak-Leibler (KL) divergence between the variational distribution and the true posterior. Thus, we are focused on solving the following optimization problem:

$$(\Gamma^*,\Delta^*,\Lambda^*) = \underset{(\Gamma,\Delta,\Lambda)}{\operatorname{argmin}}\, D(q(Z,\Theta,\Phi|\Gamma,\Delta,\Lambda)\;||\;p(Z,\Theta,\Phi|W,Y,\alpha,\eta,\beta)). \tag{A.1}$$

For the log likelihood of text corpus, we can archive its lower bound by using Jensen's inequality (Jordan et al. 1999). Take the topic modeling part for item descriptions as an example, we have:

$$\log p(w^D|\alpha_D,\eta_{D^*},\eta_{D\&A}) = \log\iiint\sum_z p\left(\phi^{D^*},\phi^{D\&A},z^D,\theta^D,w^D\;\middle|\;\alpha_D,\eta_{D^*},\eta_{D\&A}\right)d\theta^D d\phi^{D^*}d\phi^{D\&A}$$

$$= \log\iiint\sum_z \frac{p(\phi^{D^*},\phi^{D\&A},z^D,\theta^D,w^D\mid\alpha_D,\eta_{D^*},\eta_{D\&A})q(\cdot)}{q(\cdot)}d\theta^D d\phi^{D^*}d\phi^{D\&A}$$

$$= \log\underset{q(\cdot)}{\mathrm{E}}\left[\frac{p(\phi^{D^*},\phi^{D\&A},z^D,\theta^D,w^D\mid\alpha_D,\eta_{D^*},\eta_{D\&A})}{q(\cdot)}\right]$$

$$\geq \underset{q(\cdot)}{\mathrm{E}}[\log\frac{p(\phi^{D^*},\phi^{D\&A},z^D,\theta^D,w^D\mid\alpha_D,\eta_{D^*},\eta_{D\&A})}{q(\cdot)}],$$

where $q(\cdot)$ is the variational distribution.

The lower bound can be denoted by $L(\Gamma, \Delta, \Lambda; \alpha, \eta, \beta)$. Furthermore, we can find out that the difference between the log likelihood and the lower bound is exactly the KL divergence that we try to minimize in Equation (A.1).

$$p(W, Y|\alpha, \eta, \beta) = L(\Gamma, \Delta, \Lambda; \alpha, \eta, \beta) + D(q(Z, \Theta, \Phi|\Gamma, \Delta, \Lambda) \,||\, p(Z, \Theta, \Phi|W, Y, \alpha, \eta, \beta)). \tag{A.2}$$

Since the left-hand side of Equation (A.2) is irrelevant to those free variational parameters, it means that maximizing the lower bound $L(\Gamma, \Delta, \Lambda; \alpha, \eta, \beta)$ with respect to Γ, Δ and Λ is equivalent to minimizing the KL divergence between the variational posterior probability and the true posterior probability.

We can expand the lower bound by using the factorizations of p and q as below:

$$L(\Gamma, \Delta, \Lambda; \alpha, \eta, \beta) = L_{PM} + L_{TM} + H(q), \tag{A.3}$$

where L_{PM} is the log likelihood for predictive modeling, L_{TM} is the log likelihood for topic modeling, and $H(q)$ denotes the entropy of q. Note that features (f_D, f_A, f_B) in Equation (A.3.a) are actually generated from topic assignments (z^D, z^A, z^B) in Equation (A.3.b) by the element-wise product mentioned in Equation (1) in the main text. Through this process, the log likelihood for predictive modeling can be connected with the log likelihood for topic modeling. Therefore, parameters for the topic modeling component and the link prediction component can be learned simultaneously.

$$L_{PM} = L_{PM}^1 + L_{PM}^2 = \sum_{m_1, m_2} \mathbb{E}_q \begin{bmatrix} y_{m_1,m_2}^1 \log P_\sigma\left(y_{m_1,m_2}^1 \middle| f_D, f_A, f_{C,H}, \beta_D, \beta_A, \beta_{C,H}, \beta_1\right) \\ +\left(1 - y_{m_1,m_2}^1\right) \log\left(1 - P_\sigma\left(y_{m_1,m_2}^1 \middle| f_D, f_A, f_{C,H}, \beta_D, \beta_A, \beta_{C,H}, \beta_1\right)\right) \end{bmatrix}$$

$$+ \sum_{m_1, m_2} \mathbb{E}_q \begin{bmatrix} y_{m_1,m_2}^2 \log P_\sigma\left(y_{m_1,m_2}^2 \middle| f_B, f_{C,G}, \beta_B, \beta_{C,G}, \beta_2\right) \\ +\left(1 - y_{m_1,m_2}^2\right) \log\left(1 - P_\sigma\left(y_{m_1,m_2}^2 \middle| f_B, f_{C,G}, \beta_B, \beta_{C,G}, \beta_2\right)\right) \end{bmatrix}. \tag{A.3.a}$$

$$L_{TM} = \mathbb{E}_q\left[\log p\left(\phi^{D^*} \middle| \eta_{D^*}\right)\right] + \mathbb{E}_q\left[\log p\left(\phi^{D\&A} \middle| \eta_{D\&A}\right)\right] + \mathbb{E}_q\left[\log p\left(\phi^{A^*} \middle| \eta_{A^*}\right)\right] + \mathbb{E}_q\left[\log p\left(\phi^{A\&B} \middle| \eta_{A\&B}\right)\right]$$

$$+\mathbb{E}_q\left[\log p\left(\phi^{B^*} \middle| \eta_{B^*}\right)\right] + \sum_{m=1}^{M_D} \mathbb{E}_q[\log p(\theta_m^D | \alpha_D)] + \sum_{m=1}^{M_D} \sum_{n=1}^{N_m^D} \mathbb{E}_q\left[\log p\left(z_{m,n}^D \middle| \theta_m^D\right)\right]$$

$$+ \sum_{m=1}^{M_D} \sum_{n=1}^{N_m^D} \mathbb{E}_q\left[\log p\left(w_{m,n}^D \middle| \phi^{D^*}, \phi^{D\&A}, z_{m,n}^D\right)\right] + \sum_{m=1}^{M_A} \mathbb{E}_q[\log p(\theta_m^A | \alpha_A)]$$

$$+ \sum_{m=1}^{M_A} \sum_{n=1}^{N_m^A} \mathbb{E}_q\left[\log p\left(z_{m,n}^A \middle| \theta_m^A\right)\right] + \sum_{m=1}^{M_A} \sum_{n=1}^{N_m^A} \mathbb{E}_q\left[\log p\left(w_{m,n}^A \middle| \phi^{A^*}, \phi^{D\&A}, \phi^{A\&B}, z_{m,n}^A\right)\right]$$

$$+ \sum_{m=1}^{M_B} \mathbb{E}_q[\log p(\theta_m^B | \alpha_B)] + \sum_{m=1}^{M_B} \sum_{n=1}^{N_m^B} \mathbb{E}_q\left[\log p\left(z_{m,n}^B \middle| \theta_m^B\right)\right]$$

$$+ \sum_{m=1}^{M_B} \sum_{n=1}^{N_m^B} \mathbb{E}_q\left[\log p\left(w_{m,n}^B \middle| \phi^{B^*}, \phi^{A\&B}, z_{m,n}^B\right)\right]. \tag{A.3.b}$$

$$H(q) = -\mathbb{E}_q\left[\log\left(q(Z, \Theta, \Phi|\Gamma, \Delta, \Lambda)\right)\right]. \tag{A.3.c}$$

Leveraging these expanded expectations, we develop a variational EM (Expectation Maximization) algorithm. In the E(xpectation) step, we use coordinate ascent to optimize the evidence lower bound with respect to the variational parameters Γ, Δ and Λ. This yields an approximation to the true posterior. In the M(aximization) step, we employ gradient-based optimization methods to maximize the evidence lower bound for estimates of the hyper-parameters and logistic regression coefficients. The entire logic of our variational EM algorithm is described in Algorithm A.1.

Algorithm A.1: The variational EM algorithm of RTM-SHARE

```
Initialization.
repeat
    # E-step
    repeat
        for each document of item descriptions (m = 1 to M_D)
            Update δ^D, γ^D.
        for each document of guest reviews (m = 1 to M_A)
            Update δ^A, γ^A.
        for each document of host reviews (m = 1 to M_B)
            Update δ^B, γ^B.
        Update λ^{D*}, λ^{D&A}, λ^{A*}, λ^{A&B}, λ^{B*}.
    until convergence
    # M-step
    Optimize β.
until convergence.
```

In the E-step, we can obtain the updating equations for the variational parameters by computing the derivatives of the evidence lower bound and setting them to zero. Taking the iterative process for item description documents as an example, we update the variational parameters while keeping model parameters fixed according to Equation (A.4) - (A.6). The variational parameters for bilateral review documents can be updated in the same way.

1. Update δ^D:

When $t = 1, \cdots, T_{D^*}$, $\delta^D_{m,n,t} \propto \exp\left(\left[\sum_{m' \neq m} \nabla_{\delta^D_{m,n}} L^1_{PM}(m, m')\right]_t + \Psi\left(\gamma^D_{m,t}\right) - \Psi\left(\sum_{t'=1}^{T_{D^*}+T_{D\&A}} \gamma^D_{m,t'}\right) + \sum_{v=1}^{V} w_{n,v} \log \lambda^{D^*}_{t,v}\right)$;

When $t = (T_{D^*}+1), \cdots, (T_{D^*}+T_{D\&A})$, $\delta^D_{m,n,t} \propto \exp\left(\left[\sum_{m' \neq m} \nabla_{\delta^D_{m,n}} L^1_{PM}(m, m')\right]_t + \Psi\left(\gamma^D_{m,t}\right) - \Psi\left(\sum_{t'=1}^{T_{D^*}+T_{D\&A}} \gamma^D_{m,t'}\right) + \sum_{v=1}^{V} w_{n,v} \log \lambda^{D\&A}_{t-T_{D^*},v}\right)$. (A.4)

2. Update γ^D:

$$\gamma^D_{m,t} = \alpha_D + \sum_{n=1}^{N^D_m} \delta^D_{m,n,t}. \quad \text{(A.5)}$$

3. Update λ^{D^*}, $\lambda^{D\&A}$:

$$\lambda^{D^*}_{t,v} = \eta_{D^*} + \sum_{m=1}^{M_D} \sum_{n=1}^{N^D_m} \delta^D_{m,n,t} w^v_{m,n};$$

$$\lambda_{t,v}^{D\&A} = 2 * \eta_{D\&A} + \sum_{m=1}^{M_D} \sum_{n=1}^{N_m^D} \delta_{m,n,t+T_{D^*}}^{D} w_{m,n}^{v} + \sum_{m=1}^{M_A} \sum_{n=1}^{N_m^A} \delta_{m,n,t+T_{A^*}}^{A} w_{m,n}^{v}. \quad \text{(A.6)}$$

Here, $w_{m,n}^v = 1$ when $w_{m,n}$ is exactly the v-th word in the vocabulary. Besides, $\Psi(\cdot)$ is the digamma function, the first-order derivative of the log Gamma function. Similar to Chang and Blei (2010), we calculate the derivative of L_{PM}^1 by the following first-order approximation (Braun and McAuliffe 2010).

$$L_{PM}^1(m,m') = \mathbb{E}_q \begin{bmatrix} y_{m_1,m_2}^1 \log P_\sigma\left(y_{m_1,m_2}^1 \middle| f_D, f_A, f_{C,H}, \beta_D, \beta_A, \beta_{C,H}, \beta_1\right) \\ +(1 - y_{m_1,m_2}^1)\log\left(1 - P_\sigma\left(y_{m_1,m_2}^1 \middle| f_D, f_A, f_{C,H}, \beta_D, \beta_A, \beta_{C,H}, \beta_1\right)\right) \end{bmatrix}$$

$$\approx y_{m_1,m_2}^1 \log P_\sigma\left(\mathbb{E}_q\left[\beta_D f_D + \beta_A f_A + \beta_{C,H} f_{C,H} + \beta_1\right]\right) + (1 - y_{m_1,m_2}^1)\log\left(1 - P_\sigma\left(\mathbb{E}_q\left[\beta_D f_D + \beta_A f_A + \beta_{C,H} f_{C,H} + \beta_1\right]\right)\right)$$

$$= y_{m_1,m_2}^1 \log P_\sigma(\beta_D \overline{f_D} + \beta_A \overline{f_A} + \beta_{C,H} f_{C,H} + \beta_1) + (1 - y_{m_1,m_2}^1)\log\left(1 - P_\sigma\left(\beta_D \overline{f_D} + \beta_A \overline{f_A} + \beta_{C,H} f_{C,H} + \beta_1\right)\right), \quad \text{(A.7)}$$

where $\overline{f_D} = \bar{\delta}_m^D \circ \bar{\delta}_{m'}^D$, and $\bar{\delta}_m^D = \mathbb{E}_q[\bar{\mathbf{z}}_m^D] = \frac{\sum_{n=1}^{N_m^D} \delta_{m,n}^D}{N_m^D}$.

Based on Equation (A.7), the term $\nabla_{\delta_{m,n}^D} L_{PM}^1(m,m')$ in Equation (A.4) can be written as

$$\nabla_{\delta_{m,n}^D} L_{PM}^1(m,m') = \left(\nabla_{\overline{f_D}} L_{PM}^1(m,m')\right) \circ \frac{\bar{\delta}_{m'}^D}{N_m}. \quad \text{(A.8)}$$

When $y_{m,m'}^1 = 1$, the derivative in Equation (A.8) is $\nabla_{\overline{f_D}} L_{PM}^1(m,m') \approx [1 - \sigma(\beta_D \overline{f_D} + \beta_A \overline{f_A} + \beta_{C,H} f_{C,H} + \beta_1)]\beta_D$. Likewise, when $y_{m,m'}^1 = 0$, the corresponding derivative is $-\sigma(\beta_D \overline{f_D} + \beta_A \overline{f_A} + \beta_{C,H} f_{C,H} + \beta_1)\,\beta_D$.

For the M-step, we cannot obtain estimates in closed forms by setting the first-order derivatives to zero. Thus, we turn to use a gradient-based optimization method to learn the regression parameters β, which is similar to the approach adopted in the classic Logistic Rergression model. Using the approximation described in Equation (A.7), we compute the gradient of the objective given in Equation (A.3) with respect to parameters B. Because of the limited memory, we adopt the L-BFGS-B algorithm (Zhu et al. 1997) with the calculated gradient in Equation (A.9).

$$\frac{\partial L}{\partial \mathrm{B}_1} \approx \sum_{m_1,m_2}[y_{m_1,m_2}^1 - \sigma(\mathrm{B}_1 \bar{f}_1 + \beta_1)]\bar{f}_1; \frac{\partial L}{\partial \beta_1} \approx \sum_{m_1,m_2}[y_{m_1,m_2}^1 - \sigma(\mathrm{B}_1 \bar{f}_1 + \beta_1)]$$

$$\frac{\partial L}{\partial \mathrm{B}_2} \approx \sum_{m_1,m_2}[y_{m_1,m_2}^2 - \sigma(\mathrm{B}_2 \bar{f}_2 + \beta_2)]\bar{f}_2; \frac{\partial L}{\partial \beta_2} \approx \sum_{m_1,m_2}[y_{m_1,m_2}^2 - \sigma(\mathrm{B}_2 \bar{f}_2 + \beta_2)], \quad \text{(A.9)}$$

where $\mathrm{B}_1 =< \beta_D, \beta_A, \beta_{C,H} >$, $\mathrm{B}_2 =< \beta_B, \beta_{C,G} >$, $\bar{f}_1 =< \overline{f_D}, \overline{f_A}, f_{C,H} >$, $\bar{f}_2 =< \overline{f_B}, f_{C,G} >$.

With the learned model, we can predict transaction results. Given a guest and an item with its corresponding host, their matching probability in the first stage is an expectation with respect to a posterior that we cannot compute as follows

$$P\left(y_{i,j}^1 \middle| w_i^D, w_j^D, w_i^A, w_j^A, f_{C,H}\right) =$$

$$\sum_{\mathbf{z}_i^D, \mathbf{z}_j^D, \mathbf{z}_i^A, \mathbf{z}_j^A} P(y_{i,j}^1 | \bar{\mathbf{z}}_i^D, \bar{\mathbf{z}}_j^D, \bar{\mathbf{z}}_i^A, \bar{\mathbf{z}}_j^A, f_{C,H}) P(\mathbf{z}_i^D, \mathbf{z}_j^D, \mathbf{z}_i^A, \mathbf{z}_j^A | w_i^D, w_j^D, w_i^A, w_j^A). \quad \text{(A.10)}$$

Using Algorithm A.1, we can infer variational parameters which optimize the lower bound for the given evidence. Replacing the posterior with the inferred variational distribution, the predictive probability in Equation (A.10) is approximated with

$$P(y_{i,j}^1 | w_i^D, w_j^D, w_i^A, w_j^A, f_{C,H}) \approx \mathbb{E}_q[P(y_{i,j}^1 | \bar{\mathbf{z}}_i^D, \bar{\mathbf{z}}_j^D, \bar{\mathbf{z}}_i^A, \bar{\mathbf{z}}_j^A, f_{C,H})].$$

Similar logic applies to the second stage.

Appendix B. Complexity and Scalability of Learning Algorithm

To learn our topic model, we propose a variational EM algorithm, which is much less complex than exact parameter inference and convenient to speed up. In this EM algorithm (Algorithm A.1), most of the computation is used to update topic-related parameters based on item description and bilateral review documents in the E-step (Steps 4-12). Taking the iterative process for item descriptions as an example, the time complexity in Steps 5-6 is $O(M_D(T_{D^*} + T_{D\&A})V)$, where M_D is the number of item description documents (**D**), $T_{D,P}$ is the number of corpus-specific topics for item description documents, $T_{D\&A}$ is the number of shared topics for both item description documents (**D**) and guest review documents (**A**), and V is the size of the vocabulary. It is worth mentioning that the time complexity is still valid when the document length is much larger compared with the size of vocabulary. Because, we need not compute for each instance of each word in a document, but only once for each unique word in that document. Similar logic applies to the other two kinds of documents in Steps 7-8 and Steps 9-10. Based on the above, the time complexity is $O(M_D(T_{D^*} + T_{D\&A})V + M_A(T_{A^*} + T_{D\&A} + T_{A\&B})V + M_B(T_{B^*} + T_{A\&B})V)$. For Step 11, the updating here can be disassembled into three parts and processed in Steps 6, 8 and 10 respectively, and meanwhile, the total time complexity still holds.

Our proposed model has the same time computational complexity as the approach using all separate topic spaces for different documents when the number of features is the same for both approaches, namely, $T_D = T_{D^*} + T_{D\&A}$, $T_A = T_{A^*} + T_{D\&A} + T_{A\&B}$, and $T_B = T_{B^*} + T_{A\&B}$. Under this fair comparison setting, the time complexity of the approach using all separate topic spaces for different documents $O(M_D(T_D)V + M_A(T_A)V + M_B(T_B)V)$ is the same as the time complexity of the approach with both shared and corpus-specific topics $O(M_D(T_{D^*} + T_{D\&A})V + M_A(T_{A^*} + T_{D\&A} + T_{A\&B})V + M_B(T_{B^*} + T_{A\&B})V)$. Also under this setting, with the introduction of the topic space containing both shared and specific topics, the size of topic-specific word distributions is reduced from $(T_D + T_A + T_B) \times V = (T_{D^*} + 2 * T_{D\&A} + T_{A^*} + 2 * T_{A\&B} + T_{B^*}) \times V$ to $(T_{D^*} + T_{D\&A} + T_{A^*} + T_{A\&B} + T_{B^*}) \times V$. Therefore, considerable space for computation can be saved. In summary, our model based on the topic space with both shared and specific topics has the same time computational complexity as the approach using all separate topic spaces for different documents, and has lower space complexity.

As we can see, when more documents come up, the time complexity grows linearly. To alleviate the computation burden, we can implement our algorithm in parallel and distributed environments. For different kinds of documents, we can handle these kinds at the same time, because their parameters are independent when updating. Moreover, for different documents in the same kind, we can divide these documents into a few segments and process them in parallel, such as when executing Step 6 in Algorithm A.1. With the distributed computing technique, we can handle large number of documents.

Besides, our proposed model is an integrated framework and easy to scale up for new data. When some other kinds of textual information are newly introduced for more comprehensive analysis, we can directly include them in our topic model and analyze them in the same way for item descriptions and bilateral reviews. Also, when new features extracted from other sources of information like visual information come up, we can easily add them to the predictive modeling component as independent variables without any modification. We can perform feature engineering to create new features that contribute to the model based on domain knowledge of the data. In addition to feature engineering, we can easily do feature selection as well. Using a feature selection technique, we can filter out those redundant or irrelevant features and obtain a simplified model.

Appendix C. Simulation for Topic Space

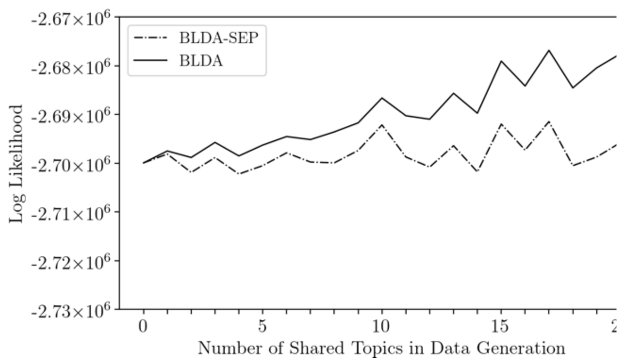
To illustrate the benefit of using the topic structure with shared and corpus-specific topics, we apply the designed algorithm to simulated data for theoretical analysis. More specifically, we generate two types of documents by varying the degree of shared topics and examine the performance between our model (BRTM) which considers both shared and corpus-specific topics and a benchmark model (BRTM-SEP) where the two types of documents are processed separately using two distinct LDA models. In the simulation, we exclude the regression prediction component by focusing explicitly on the topic generation task, therefore we just use LDA instead of RTM, and call these two models BLDA and BLDA-SEP. This allows us to isolate the issue of evaluating the quality of the topics generated without considering how the topics are used for other purposes such as prediction.

We generate the simulated data according to the following procedure. Two sets of documents were generated based on a vocabulary of 1000 distinct words. We assume that each set of documents has 20 topics. We vary the number of shared topics from 0 to 20 and simultaneously the number of corpus-specific topics for each set of documents decreases from 20 to 0. When the number of shared topics is fixed, a respective dataset is generated based on the generative process described in Parts II, III and IV of the Entire Generative Process in Section 4.5 in the main text. We obtain topic-specific word distributions of corpus-specific and shared topics. For each set of documents, 1000 documents are generated, and each one of them contains 200 words. Based on the generated data, we examine two topic models: BLDA and BLDA-SEP. Since the words in the vocabulary in the simulation are all number indexes without actual meanings, we adopt the log-likelihood measure to compare the two models.

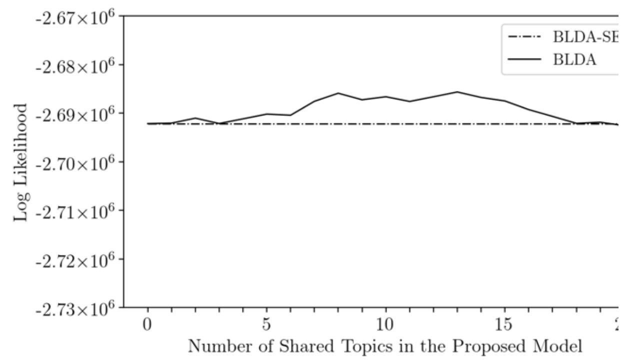
Figure C.1(a) presents the first set of simulation results. As the number of shared topics varies from 0 to 20, we generated 21 different datasets. For each of the 21 datasets generated, we compare the two candidate topic models (BLDA and BLDA-SEP). The number of shared topics used in BLDA is exactly the same as the one used when generating the data. This enables us to observe the best-case scenario, and see how much benefit we can potentially obtain if we set the right number of shared topics in our model. As shown in Figure C.1(a), the performance gap widens between the two models as the number of shared topics increases. Figure C.1(b) presents another set of simulation results. In this simulation setting, we use one generated dataset with 10 shared topics. Then we vary the number of shared topics parameter in the BLDA model to see how the performance changes compared to that of the BLDA-SEP model. As the number of shared topics deviates more from the true value 10, although the performance advantage over BLDA-SEP shrinks, BLDA performs better for most cases. This suggests that in real settings where we do not know the true value for the number of shared topics, the BLDA model can still outperform the BLDA-SEP model when the number of shared topics is set at a reasonable value. Another thing that is worthwhile mentioning is that the number of model parameters decreases with the introduction of shared topics. This is because one shared topic can substitute two corpus-specific topics. Normally, as the number of model parameters decreases, the log-likelihood of the model tends to decrease as well. However, the log-likelihood of our proposed model is still better than that of the original model further accentuating the superiority of our model.

Figure C.1. Effects of introducing the topic space with shared topics.

(a) Multiple datasets, BLDA with the correct number of shared topics.



(b) Single dataset, BLDA with different number of shared topics.



Appendix D. A Detailed Description of Host Reviews

On Airbnb, hosts sometimes leave reviews with similar contents for different transactions, such as copying short sentences with generic information or slightly changing the name of the guest. The similar reviews written by the same host are called redundant reviews. When generating the review collections used in our models, we only keep one review for each group of highly similar reviews by the same host and remove the redundant ones.

For each host review, we use an open source toolkit for nature language processing named TextBlob (Loria et al. 2018) to tokenize the text into words and then label each word with its part-of-speech tag. For each tokenized word, we apply a wordnet lemmatizer (Miller 1998) to reduce the inflected forms of the word. Afterward, we remove stopwords[1] and common names[2]. Through these text preprocessing procedures, each review can be transformed into a set of words. If two reviews are mapped into the same set of words, we can identify these two reviews as redundant reviews. We only keep one copy for each group of highly similar reviews and remove the other redundant reviews. As summarized in Table D.1, 34,416 hosts in our NYC dataset have written 836,044 reviews, among which 813,429 are written in English. For these reviews in English, 19.5% of them are similar to some other reviews (at least one) written by the same host in contents. Since we only keep one copy for each group of redundant reviews written by the same host, 14.3% of host reviews are filtered.

Table D.1. Detailed statistics for redundant reviews in host reviews.

# of hosts	# of listings	# of reviews	# of reviews (in English)
34,416	43,395	836,044	813,429
# of redundant reviews	**% of redundant reviews**	**# of unique reviews**	**% of filtered reviews**
158,541	19.5%	696,711	14.3%

Next, we demonstrate that there exists significant topic variance among the topics we discover after removing the redundant reviews, reflecting the fact that host reviews do contain a variety of aspects about the transactions and guests. These perspectives can be well captured by topic modeling. In order to show topic variance in host reviews, we apply Latent Dirichlet Allocation (Blei et al. 2003) on host reviews and extracted many meaningful topics. We set the number of topics to 50, and Figure D.1 below illustrates all the 50 learned topics. Each box corresponds to a topic, and the bigger the word, the more representative the word is in that topic. As we can see by glancing the 50 boxes below, the main representative words for these topics are quite different showing that there is significant topic variance among host reviews.

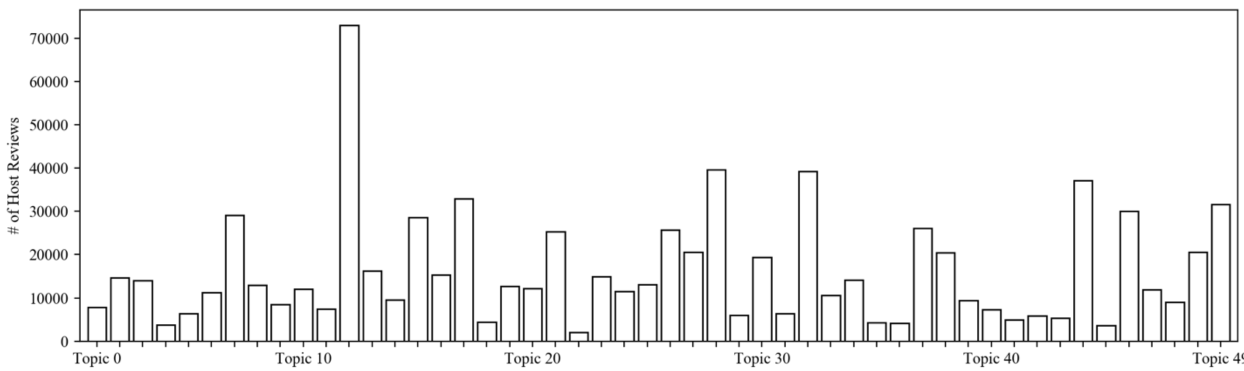
[1] The stopword list is obtained from https://www.ranks.nl/stopwords

[2] Frequently occurring names from Census 1990, https://catalog.data.gov/dataset/names-from-census-1990

Figure D.1. The representative words of each topic learned from host reviews.

In addition to show that the topics within host reviews are different, we also show that all the 50 topics have enough representation among all the host reviews. For each host review, we represent it as a topic vector, and identify the topic with the highest value in this vector as the representative topic for this review. Figure D.2 below shows the number of host reviews with a given topic as its representative topic. As shown in Figure D.2, the number of host reviews represented by each topic does vary but is not that extreme. There are 31 topics (out of 50) represent more than 10,000 host reviews, and about 90% (44 of 50) topics represent more than 5,000 host reviews. This topic analysis indicates a wide topic variance among host reviews.

Figure D.2. The number of host reviews represented by each topic.



Appendix E. How to Handle Missing Negative Instances

In the main text, we presented two ways to handle missing negative instances (BRTM-Rank and BRTM-Sample). Section E.1 presents more details about BRTM-Rank which is based on a learning-to-rank model where the positive instances are considered ranked higher than the unobserved instances. Section E.2 presents more details about BRTM-Sample which is to generate negative instances by sampling unobserved instances. In Section E.3, we provide more discussions about alternative solutions to handle missing negative instances.

E.1 Details of BRTM-Rank

In BRTM-Rank, we use a pairwise ranking loss function named Bayesian Personalized Ranking (BPR) (Rendle et al. 2009) rather than the loss function of logistic regression. In the BPR framework, positive feedback is assumed to be preferable than unobserved feedback. We introduce this assumption into our proposed model to alleviate the issue of lacking negative instances.

In our scenario, the probability of booking the listing in the observed transaction is assumed higher than the probability of booking other listings. For each observed transaction in the training set involving guest g_i, host h_{k^+}, and listing l_{j^+}, the probability of this transaction being successful is denoted as $P_\sigma(Y_{i,j^+,k^+} = 1) = \sigma(\beta_D f_D^+ + \beta_A f_A^+ + \beta_{C,H} f_{C,H}^+ + \beta_B f_B^+ + \beta_{C,G} f_{C,G}^+ + \beta_0)$. Note that we use "+" to represent the observed transaction, and use "−" to indicate the unobserved transactions. Given a listing l_{j^-}, which is active at the time of the observed transaction but not actually booked, the probability that guest g_i has a transaction on listing l_{j^-} with the corresponding host h_{k^-} is $P_\sigma(Y_{i,j^-,k^-} = 1) = \sigma(\beta_D f_D^- + \beta_A f_A^- + \beta_{C,H} f_{C,H}^- + \beta_B f_B^- + \beta_{C,G} f_{C,G}^- + \beta_0)$. We can pair each observed transaction with everyone among the unobserved transactions and generate multiple pairs ($S = \{(i, j^+, k^+, j^-, k^-)\}$).

With the defined probability, we analyze the pairwise preference in each pair and calculate the BPR loss for optimization as $\sum_{(i,j^+,k^+,j^-,k^-)\in S}[\ln \sigma(P_\sigma(Y_{i,j^+,k^+} = 1) - P_\sigma(Y_{i,j^-,k^-} = 1))]$. Thus, the term for predictive modeling in the evidence lower bound (Equation (A.3.a) in Appendix A) can be updated as

$$L_{PM} = \sum_{(i,j^+,k^+,j^-,k^-)\in S} \mathbb{E}_q[\ln \sigma\left(P_\sigma(Y_{i,j^+,k^+} = 1) - P_\sigma(Y_{i,j^-,k^-} = 1)\right)]$$

Accordingly, the derivative in Equation (A.8) of Appendix A is rewritten as

$$\nabla_{\overline{f_{D^+}}} L_{PM}(m, m') \approx \left(1 - \sigma\left(P_\sigma(Y_{i,j^+,k^+} = 1) - P_\sigma(Y_{i,j^-,k^-} = 1)\right)\right) P_\sigma(Y_{i,j^+,k^+} = 1)(1 - P_\sigma(Y_{i,j^+,k^+} = 1))\beta_D;$$

$$\nabla_{\overline{f_{D^-}}} L_{PM}(m, m') \approx \left(1 - \sigma\left(P_\sigma(Y_{i,j^+,k^+} = 1) - P_\sigma(Y_{i,j^-,k^-} = 1)\right)\right) (-P_\sigma(Y_{i,j^-,k^-} = 1))(1 - P_\sigma(Y_{i,j^-,k^-} = 1))\beta_D.$$

The rest of the learning process is similar to that in Appendix A.

E.2 Details of BRTM-Sample

In BRTM-Sample, we treat all the unobserved instances as negative and sample several instances for training. We randomly select negative instances from listings active at the time of the transaction and similar to the actually booked listing which means that these listings are more likely to satisfy the guest's search criteria and be shown to the guest. Instead of consider all the active listings, under BRTM-Sample, negative instances are selected from a smaller consideration set which contains listings more similar to the unobserved ones shown to the guest before booking. The listings tailored to the guest' search needs are more likely to be the ones presented but not booked, which can be a good proxy for negative instances. Specifically, we randomly select negative instances from active listings which are similar to the booked listing in terms of three basic search filters: the room type (exactly the same), price (ranging from -50% to +50%), and location (within two kilometers).

This unobserved-as-negative strategy is also adopted in recommendation systems research when the feedback data is not sufficient (Hu et al. 2008, Pan et al. 2008, Pan and Scholz 2009, He et al. 2016). Since the number of negative instances is far more than that of positive ones, we adopt a widely-used sampling technique (He and Garcia 2008) to balance the ratio of negative instances to positive ones.

E.3 Other Alternative strategies of Handling Missing Negative Instances

In addition to the above two solutions we presented in the main text, there are other possible strategies to handle missing negative instances. Since we think BRTM-Rank and BRTM-Sample are the most appropriate solution for our context, we discuss other alternative ways here in this Appendix.

The first alternative strategy (BRTM-Positive) is to modify our proposed model slightly so that it does not require negative instances. The second alternative strategy (BRTM-Sentiment) also only uses the observed instances by adopting a slightly different way to define positive and negative instances.

E.3.1 Details of BRTM-Positive

In BRTM-Positive, we modify our proposed model slightly so that it can be implemented with only positive instances. Instead of the logistic link function, we use a probabilistic linear model with normal observation noise to model transaction status, which is similar to the Probabilistic Matrix Factorization model (Mnih and Salakhutdinov 2008). In specific, the original link function in Equation (2) in the main text is modified as

$$P(Y_{i,j,k} \mid \beta, f, \sigma^2) = N(Y_{i,j,k} \mid \beta_D f_D + \beta_A f_A + \beta_{C,H} f_{C,H} + \beta_B f_B + \beta_{C,G} f_{C,G},\ \sigma^2),$$

where $N(\cdot|\mu,\sigma^2)$ is the probability density function of the normal distribution with mean μ and variance σ^2. Thus, results of the observed transactions ($Y_{i,j,k}$) can be fitted with the linear combination of feature vectors ($\beta_D f_D + \beta_A f_A + \beta_{C,H} f_{C,H} + \beta_B f_B + \beta_{C,G} f_{C,G}$).

With this modification, the learning process of this model is as follows. The previous log likelihood for predictive modeling in Equation (A.3.a) in Appendix A can be updated as

$$L_{PM} = \mathbb{E}_q\left[\log P\left(\beta_D f_D + \beta_A f_A + \beta_{C,H} f_{C,H} + \beta_B f_B + \beta_{C,G} f_{C,G} \middle| Y_{i,j,k}, \sigma^2\right)\right].$$

Here, maximizing the log-likelihood term is equivalent to minimizing the sum of the squared error between $Y_{i,j,k}$ and $\beta_D f_D + \beta_A f_A + \beta_{C,H} f_{C,H} + \beta_B f_B + \beta_{C,G} f_{C,G}$. Thus, L_{PM} can be expanded as $\sum_{m_1,m_2} \mathbb{E}_q\left[-\frac{1}{2}(y_{m_1,m_2} - \beta_D f_D - \beta_A f_A - \beta_{C,H} f_{C,H} - \beta_B f_B - \beta_{C,G} f_{C,G})^2\right]$. Accordingly, the derivative in Equation (A.8) in Appendix A is rewritten as $\nabla_{\overline{f_D}} L_{PM}(m,m') \approx (y_{m_1,m_2} - \beta_D f_D - \beta_A f_A - \beta_{C,H} f_{C,H} - \beta_B f_B - \beta_{C,G} f_{C,G})\beta_D$. The rest of the learning process is similar to that presented in Appendix A.

E.3.2 Details of BRTM-Sentiment

In BRTM-Sentiment, we label observed transactions based on sentiment analysis of bilateral reviews. For each observed transaction, the sentiment of the guest's review is used as the label in Stage 1 of our proposed model, and the sentiment of the host's review is used as the label in Stage 2. In this way, we can directly implement our model without making any modifications in Section 5.2.1.

For sentiment analysis, we apply a well-trained Naïve Bayes classifier provided by a widely-used open source toolkit for nature language processing named TextBlob (Loria et al. 2018). The sentiment score is a numerical value between zero and one. To directly test the validity of our proposed model which requires transaction status in both stages, we use a cut-off sentiment score 0.5 to label observed transactions into positive and negative instances.

Note that the Naïve Bayes classifier (NaiveBayesAnalyzer[3] in TextBlob) used in the implementation of BRTM-Sentiment is already trained using their labeled dataset, and we just apply that trained classifier directly without training it ourselves. In our study, we use this trained classifier to generate labels for our model.

[3] Source code: https://textblob.readthedocs.io/en/dev/_modules/textblob/en/sentiments.html

Appendix F. Baselines

F.1 CF-G

For Collaborative Filtering based baselines, we tailor the classic item-based Collaborative Filtering (Sarwar et al. 2001) to incorporate text contents. Taking CF-G as an example, for a given guest and a candidate listing, we calculate the similarity between the candidate listing and each historical listing that the guest has ever reviewed before, and then take an average over all the historical listings. The similarity is calculated by using the Jaccard similarity between two collections of guest reviews: (a) all the reviews received by the historical listing, and (b) all the reviews about the candidate listing. According to the calculated average similarity, we can rank all candidate listings.

F.2 LDA-G

For the LDA-based models, we obtain topic vectors for all the documents by LDA, use these vectors to calculate the similarity between documents, and rank candidate listings based on the similarity. Taking LDA-G as an example, for a given guest and a candidate listing, we calculate the cosine similarity between the topic vector of the document containing all the reviews posted by the guest and the topic vector of the document containing all the reviews written for the listing.

F.3 RTM-G, RTM-GH, BRTM-SEP

In order to show the advantages of our proposed BRTM model designed to handle bilateral reviews, we compare with the model treating bilateral reviews as unilateral reviews by not differentiating guest reviews and host reviews (RTM-GH). In RTM-GH, guest reviews and host reviews are mixed together and share a single topic space. We also compare with the model handling bilateral reviews separately without considering shared topics (BRTM-SEP) in order to illustrate the benefit of our topic structure with both shared and corpus-specific topics. These two baseline models also consider item descriptions and user profiles. Please see Appendix G (under Section G.1) for how the features spaces of these three models (RTM-GH, BRTM-SEP, and BRTM) differ, and explanations for why BRTM is superior. Compared to RTM-GH, RTM-G only uses guest reviews.

F.4 Implementation of the Supervised Topic Labeling Model (STL)

We design a supervised topic labeling model (STL) based on Abrahams et al. (2012) to manually summarize major topics, label a subset of documents, and build classifiers for topic assignment. With the supervised topic representations, we then implement a Logistic Regression component for transaction success prediction.

First of all, we use Amazon MTurk to identify major topics from a subset of listing descriptions, guest reviews, and host reviews. For each document, the MTurk master worker is required to summarize it using key words or phrases. Each document is summarized by at least two MTurk master workers. For each of the three text input types, we obtain 20 major topics based on the most frequently mentioned key words. Next, we recruit two experts to independently code 100 listing descriptions, 100 guest reviews, and 100 host reviews with the identified topics. As displayed in Table F.1, every identified topic has a decent inter-rater reliability (the Cohen's Kappa is greater than 0.6), which means substantial agreements. Table F.1 also shows the number of reviews containing each topic identified by each expert (# of positive instances) and the number of agreements.

Table F.1. The major topics identified from listing descriptions, guest reviews, and host reviews.

(a) From listing descriptions.

Topic	# of positive instances		# of agreements	Inter-rater reliability (Kappa)
	By expert 1	By expert 2		
"public transportation"	78	84	76	0.68
"bedroom"	52	54	44	0.64
"kitchen"	51	49	43	0.72
"neighborhood"	29	37	27	0.73
"close to"	36	30	26	0.68
"bathroom"	38	27	25	0.66
"location"	24	34	21	0.62
"spacious"	21	26	20	0.81
"wifi"	27	22	19	0.70
"host is available"	28	24	19	0.64
"shopping"	31	22	19	0.62
"television"	22	18	16	0.75
"private space"	21	21	16	0.70
"comfortable place"	18	23	15	0.66
"convenience"	15	13	10	0.67
"furnished"	11	13	9	0.72
"view"	8	7	7	0.93
"price"	7	7	6	0.85
"renovated"	7	8	6	0.78
"parking"	8	8	6	0.73

(b) From guest reviews.

Topic	# of positive instances		# of agreements	Inter-rater reliability (Kappa)
	By expert 1	By expert 2		
"good location"	58	61	53	0.73
"clean"	30	32	30	0.95
"helpful"	28	34	27	0.81

"public transportation"	24	31	22	0.73
"feel comfortable"	20	23	18	0.79
"neighborhood"	19	19	17	0.87
"responsive"	16	17	15	0.89
"friendly"	15	21	15	0.80
"communication"	13	15	13	0.92
"safety"	12	14	12	0.91
"kitchen"	12	11	11	0.95
"spacious"	10	16	10	0.74
"food"	15	12	10	0.70
"quiet"	9	11	9	0.89
"welcoming"	10	11	9	0.84
"bathroom"	11	11	9	0.80
"bedroom"	11	8	7	0.71
"air condition"	5	5	5	1.0
"wifi"	5	5	5	1.0
"television"	4	4	4	1.0

(c) From host reviews.

Topic	# of positive instances		# of agreements	Inter-rater reliability (Kappa)
	By expert 1	By expert 2		
"nice"	60	64	56	0.75
"communication"	35	37	34	0.91
"clean"	33	34	32	0.93
"group or family"	33	32	29	0.84
"highly recommend"	23	27	22	0.84
"respectful"	20	25	20	0.86
"friendly"	19	26	19	0.8
"tidy"	15	22	15	0.77
"quiet"	9	9	9	1
"follow house rule"	6	7	6	0.92
"considerate"	7	6	5	0.75
"check-in and check-out"	5	4	4	0.88
"polite"	3	3	3	1
"responsive"	3	3	3	1
"responsible"	3	3	3	1
"feel comfortable"	3	4	3	0.85
"cooperative"	2	2	2	1
"organized"	2	3	2	0.8

"independent"	4	2	2	0.66
"community"	2	1	1	0.66

Afterwards, two experts are asked to resolve their remaining differences. Table F.2 summarizes the final number of positive instances for each topic after the consolidation together with the Kappa values. Both Tables F.1 and F.2 show that the more general topics appear in more reviews and those more specific topics appear in fewer reviews.

Table F.2. The Kappa and number of positive instances for all major topics identified.

From listing descriptions			From guest reviews			From host reviews		
Topic	**Kappa**	**# of pos. instances**	**Topic**	**Kappa**	**# of pos. instances**	**Topic**	**Kappa**	**# of pos. instances**
"public transportation"	0.68	86	"good location"	0.73	66	"nice"	0.75	68
"bedroom"	0.64	62	"helpful"	0.81	35	"communication"	0.91	38
"kitchen"	0.72	57	"public transportation"	0.73	33	"group or family"	0.84	36
"close to"	0.68	40	"clean"	0.95	32	"clean"	0.93	35
"bathroom"	0.66	40	"feel comfortable"	0.79	25	"highly recommend"	0.84	28
"neighborhood"	0.73	39	"neighborhood"	0.87	21	"friendly"	0.80	26
"location"	0.62	37	"friendly"	0.8	21	"respectful"	0.86	25
"shopping"	0.62	34	"responsive"	0.89	18	"tidy"	0.77	22
"host is available"	0.64	33	"food"	0.70	17	"quiet"	1.0	9
"wifi"	0.70	30	"spacious"	0.74	16	"considerate"	0.75	8
"spacious"	0.81	27	"communication"	0.92	15	"follow house rule"	0.90	7
"private space"	0.70	26	"safety"	0.91	14	"check-in and check-out"	0.88	5
"comfortable place"	0.66	26	"bathroom"	0.80	13	"responsive"	1.0	4
"television"	0.75	24	"kitchen"	0.95	12	"feel comfortable"	0.85	4
"convenience"	0.67	18	"welcoming"	0.84	12	"responsible"	1.0	3
"furnished"	0.72	15	"bedroom"	0.71	12	"organized"	0.80	3
"parking"	0.73	10	"quiet"	0.89	11	"independent"	0.66	3
"renovated"	0.78	9	"air condition"	1.0	5	"polite"	1.0	2
"view"	0.93	8	"wifi"	1.0	5	"cooperative"	1.0	2
"price"	0.85	8	"television"	1.0	4	"community"	0.66	2

With this labeled dataset, we train a binary classifier using Logistic Regression for each major topic, and automatically label all the other documents in our data. In this way, each document is represented by a topic vector with 20 dimensions. Finally, we implement a Logistic Regression component for transaction success prediction and summarize the results in Table F.3. In addition, we examine all the identified topics/features and find that a majority of them are included in topics/features learned by our BRTM model, as illustrated in the last column of Table F.3.

Table F.3. Summary of Logistic Regression results by STL.

#	Feature in STL	Est.	Std. Error	Sig.	Topic by BRTM-Sample
1	"close to", from *D*	0.093	0.06	0.11	"close to park", #32 of corpus-specific for *D*
2	"kitchen", from *D*	0.071	0.05	0.18	"room situation", #9 of shared by *D* & *A*
3	"convenience", from *D*	-0.058	1.10	0.96	"walking minutes to station", #24 of corpus-specific for *D*
4	"price", from *D*	-0.57	2.27	0.80	"check-in policy", #26 of corpus-specific for *D*
5	"renovated", from *D*	-0.16	0.30	0.58	/
6	"parking", from *D*	-0.79	1.33	0.55	"city centre & free parking", #29 of corpus-specific for *D*
7	"host is available", from *D*	0.29	0.05	***	/
8	"private space", from *D*	0.50	0.16	**	"private room", #11 of corpus-specific for *D*
9	"view", from *D*	9.81	3.53	*	"view", #14 of shared by *D* & *A*
10	"comfortable place", from *D*	0.31	0.36	0.39	"clean & nice", #1 of shared by *D* & *A*
11	"furnished", from *D*	-6.41	2.81	*	/
12	"neighborhood", from *D*	-0.21	0.17	0.22	"block & neighborhood", #4 of shared by *D* & *A*
13	"wifi", from *D*	0.13	0.10	0.18	/
14	"public transportation", from *D*	3.79	0.06	0	"airport transportation", #18 of corpus-specific for *D*
15	"bedroom", from *D*	0.21	0.04	***	"bedroom suite", #29 of corpus-specific for *D*
16	"bathroom", from *D*	0.093	0.06	0.11	"private room", #11 of corpus-specific for *D*
17	"television", from *D*	0.024	0.21	0.91	/
18	"location", from *D*	0.23	0.10	*	"great location", #10 of shared by *D* & *A*
19	"spacious", from *D*	-0.17	0.07	*	"large area", #3 of corpus-specific for *D*
20	"shopping", from *D*	0.15	0.05	*	/
21	"good location", from *A*	0.81	0.03	0	"great location", #15 of corpus-specific for *A*
22	"feel comfortable", from *A*	0.59	0.04	0	"comfortable", #2 of corpus-specific for *A*
23	"communication", from *A*	0.39	0.08	***	"smooth communication", #3 of shared by *A* & *B*
24	"spacious", from *A*	0.017	0.15	0.91	/
25	"clean", from *A*	0.42	0.04	0	"super clean", #9 of shared by *A* & *B*
26	"bathroom", from *A*	1.16	3.32	0.73	"clean room", #19 of specific for *A*
27	"quiet", from *A*	0.16	0.51	0.75	/
28	"kitchen", from *A*	-3.38	1.94	.	"room situation", #9 of shared by *D* & *A*
29	"food", from *A*	-2.47	1.23	*	"breakfast", #20 of shared by *D* & *A*
30	"television", from *A*	4.83E-06	0.00	0.58	/
31	"neighborhood", from *A*	-0.54	0.23	*	"block & neighborhood", #4 of shared by *D* & *A*

32	"responsive", from *A*	-0.75	0.95	0.43	"smooth communication", #3 of shared by *A* & *B*
33	"air condition", from *A*	-4.85E-03	0.01	0.41	/
34	"bedroom", from *A*	-11.63	3.51	**	"room situation", #9 of shared by *D* & *A*
35	"transportation", from *A*	0.39	0.05	***	"subway", #17 of shared by *D* & *A*
36	"welcoming", from *A*	-3.13	2.21	0.16	"great experience", #5 of shared by *A* & *B*
37	"wifi", from *A*	0.25	1.11	0.82	/
38	"friendly", from *A*	0.073	0.08	0.36	"great experience", #5 of shared by *A* & *B*
39	"helpful", from *A*	0.43	0.07	***	"helpful & comfortable", #4 of shared by *A* & *B*
40	"safety", from *A*	-2.53	1.40	.	/
41	"communication", from *B*	-0.081	0.07	0.22	"smooth communication", #3 of shared by *A* & *B*
42	"respectful", from *B*	0.26	0.07	***	"respectful", #49 of corpus-specific for *B*
43	"clean", from *B*	0.049	0.04	0.20	"super clean", #9 of shared by *A* & *B*
44	"follow house rule", from *B*	3.03	3.52	0.39	"follow house rules", #15 of corpus-specific for *B*
45	"polite", from *B*	0.014	0.21	0.95	"polite & friendly", #6 of corpus-specific for *B*
46	"nice", from *B*	0.31	0.03	***	"nice & clean", #47 of corpus-specific for *B*
47	"highly recommend", from *B*	0.44	0.04	***	"highly recommend", #1 of corpus-specific for *B*
48	"feel comfortable", from *B*	0.19	0.70	0.79	"helpful & comfortable", #4 of shared by *A* & *B*
49	"responsive", from *B*	5.31E-04	0.00	0.54	"responsible", #45 of corpus-specific for *B*
50	"friendly", from *B*	0.37	0.07	***	"friendly", #20 of corpus-specific for *B*
51	"considerate", from *B*	0.28	1.07	0.79	"considerate & friendly", #37 of corpus-specific for *B*
52	"cooperative", from *B*	1.58E-05	0.00	0.65	"good to deal with", #7 of corpus-specific for *B*
53	"group or family", from *B*	0.17	0.06	**	"family & friend", #2 of corpus-specific for *B*
54	"organized", from *B*	1.03	0.96	0.28	/
55	"quiet", from *B*	0.49	0.11	***	"quiet", #51 of corpus-specific for *B*
56	"community", from *B*	1.74	3.27	0.60	"neighborhood & family", #7 of shared by *A* & *B*
57	"tidy", from *B*	0.16	0.04	***	"tidy", #41 of corpus-specific for B
58	"responsible", from *B*	0.11	0.19	0.55	"responsible", #45 of corpus-specific for *B*
59	"independent", from *B*	0.55	2.55	0.83	"independent & quiet", #4 of corpus-specific for *B*
60	"check-in and check-out", from *B*	0.30	1.08	0.78	"check-out", #8 of corpus-specific for *B*

F.5 Implementation of the Deep Learning Baseline

In our experiments, we implement two representative deep learning models as baselines: DL-CNN and DL-MLP. DL-CNN is based on Convolutional Neural Networks (Krizhevsky et al. 2012), and DL-MLP is based on Multi-Layer Perceptron (Minsky and Papert 1969). Both models are effective in text classification (Collobert 2011, Kim 2014, Zhang et al. 2018). In order to evaluate the effectiveness of our feature space, these two deep learning baselines are directly applied to documents without built on top of the proposed feature space, which is our main contribution. Details of how to implement DL-CNN and DL-MLP are respectively provided in Section F.5.1 and F.5.2, and the experimental results comparing with these baselines are displayed in Appendix I.

F.5.1 Details of how to implement DL-CNN

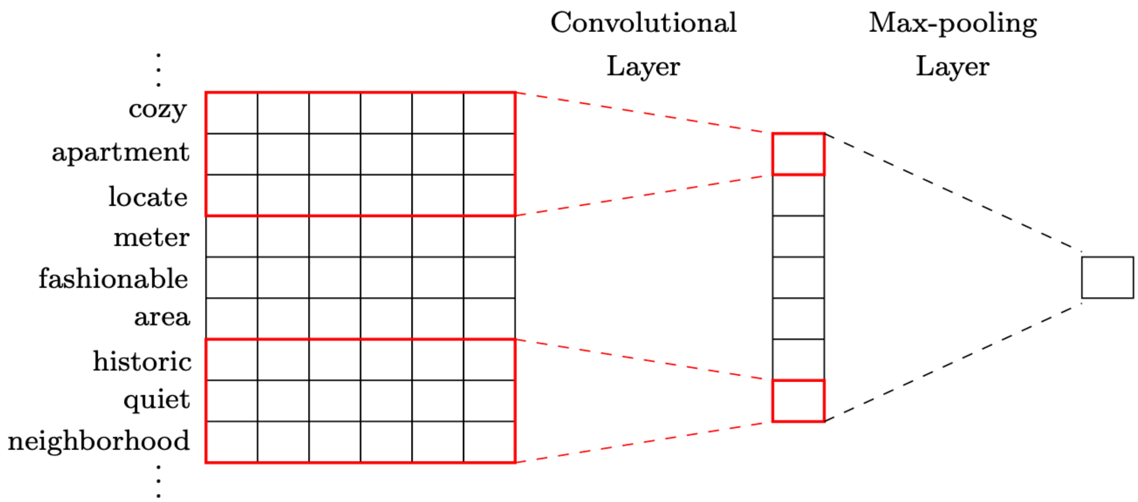
When implementing the DL-CNN baseline, we follow the idea of using CNNs to classify documents in Kim (2014). First, we represent documents as matrices through word embedding techniques (Mikolov et al. 2013a & 2013b, Pennington et al. 2014) where words are mapped into real-valued vectors. Each document used in our predictive framework can be regarded as a sequence of words, and each word in the document corresponds to a real-valued vector. Thus, each document can be represented as a two-dimensional matrix. The size of the first dimension is the number of words in the document, and the size of the second dimension is the length of the word embedding. As displayed in Table F.4, texts (D_{L_i}, D_j, $A_{(i,\cdot)}$, $A_{(\cdot,j)}$), $B_{(\cdot,i)}$, and $B_{(k,\cdot)}$) are represented respectively by sequences of word embeddings (e_i^D, e_j^D, e_i^A, e_j^A, e_i^B, and e_k^B). In our implementation, word embeddings are learned from all three types of documents together with predictive modeling. Since word embeddings are usually around 100 and 300 in length (Mikolov et al. 2013a & 2013b, Pennington et al. 2014), we vary this parameter and pick the value with the best predictive performance, which is 300.

Table F.4. Summary of text representations in DL-CNN.

Texts	Collection of Listing Descriptions		Collection of Guest Reviews		Collection of Host Reviews	
	D_{L_i}	D_j	$A_{(i,\cdot)}$	$A_{(\cdot,j)}$	$B_{(\cdot,i)}$	$B_{(k,\cdot)}$
Sequences of Word Embeddings	e_i^D	e_j^D	e_i^A	e_j^A	e_i^B	e_k^B

With these sequences of word embeddings, we then apply the Convolutional Neural Network to extract features for prediction. Following the model proposed by Kim (2014), we adopt horizonal convolutional filters to extract features from the sequences of word embeddings. The model architecture of our applied CNN is presented in Figure F.1.

Figure F.1. Model architecture of our applied CNN.



As shown in Figure F.1, each horizontal filter has a fixed width, which is the same as the size of the word embedding, and a varied length, such as three word positions. The filter can be regarded as a window sliding over the matrix (the sequence of word embeddings) from top to bottom. As the convolutional layer is applied to each possible window of words, a feature map can be generated. We subsequently perform a max-pooling layer on the feature map to capture the most important feature generated by the convolutional layer. As summarized in Table F.5, we vary the length of the filter from three to five word positions and respectively extract features from texts. Taking the collection of listing descriptions as an example, we extract features f_i^D-3, f_i^D-4, and f_i^D-5 (or f_j^D-3, f_j^D-4, and f_j^D-5) from D_{L_i} (or D_j).

Table F.5. Summary of features extracted in DL-CNN.

Texts	Collection of Listing Descriptions		Collection of Guest Reviews		Collection of Host Reviews	
	D_{L_i}	D_j	$A_{(i,\cdot)}$	$A_{(\cdot,j)}$	$B_{(\cdot,i)}$	$B_{(k,\cdot)}$
Sequences of Word Embeddings	e_i^D	e_j^D	e_i^A	e_j^A	e_i^B	e_k^B
Extracted Features	f_i^D-3, f_i^D-4 & f_i^D-5	f_j^D-3, f_j^D-4 & f_j^D-5	f_i^A-3, f_i^A-4 & f_i^A-5	f_j^A-3, f_j^A-4 & f_j^A-5	f_i^B-3, f_i^B-4 & f_i^B-5	f_k^B-3, f_k^B-4 & f_k^B-5

After concatenating all the extracted features and other profile-based features, we can connect them to the transaction results by the logistic link function.

F.5.2 Details of how to implement DL-MLP

In DL-MLP, we first represent text inputs as vectors through the Vector Space Model (Salton et al. 1975). As summarized in Table F.6, constructed documents used in our predictive framework

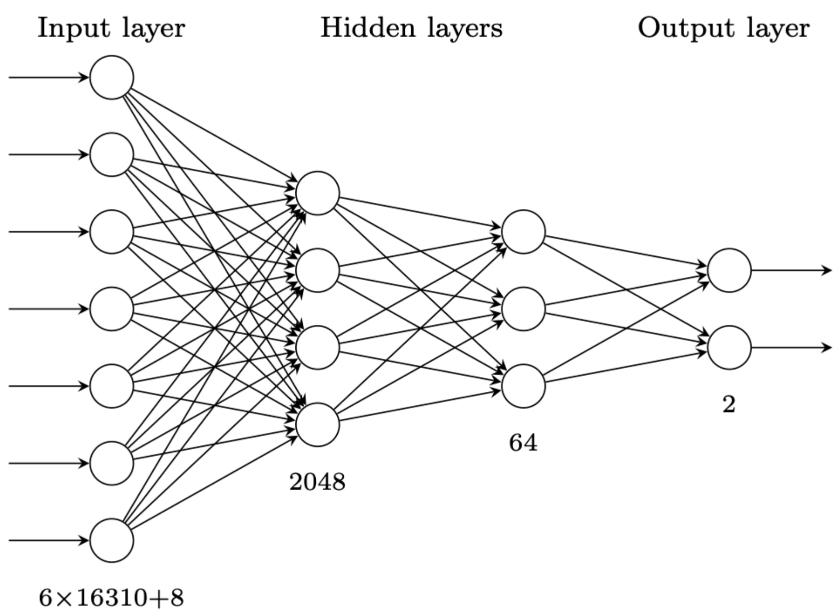
(D_{L_i}, D_j, $A_{(i,\cdot)}$, $A_{(\cdot,j)}$), $B_{(\cdot,i)}$, and $B_{(k,\cdot)}$) are represented respectively by vectors (v_i^D, v_j^D, v_i^A, v_j^A, v_i^B, and v_k^B). In each vector, each dimension corresponds to a word, so the length of the vector is the size of the word vocabulary. For each document and its corresponding vector, if a word occurs in the document, the corresponding element in the vector has a non-zero weight. Following the classic Vector Space Model proposed by Salton et al. (1975), we use TF-IDF (Term Frequency-Inverse Document Frequency) to calculate the weights.

Table F.6. Summary of vectors used in DL-MLP.

Texts	Collection of Listing Descriptions		Collection of Guest Reviews		Collection of Host Reviews	
	D_{L_i}	D_j	$A_{(i,\cdot)}$	$A_{(\cdot,j)}$	$B_{(\cdot,i)}$	$B_{(k,\cdot)}$
Vectors	v_i^D	v_j^D	v_i^A	v_j^A	v_i^B	v_k^B

With the obtained vectors and other profile-based features, we then apply a Multi-Layer Perceptron network (Minsky and Papert 1969) for prediction. As shown in Figure F.2, there are four layers in our applied MLP: one input layer, two hidden layers, and one output layer. All the obtained vectors and other profile-based features are fed into the input layer, and then passed through two hidden layers with non-linearly activating nodes, finally mapped into the output layer with a logistic link function. All these four layers are fully connected.

Figure F.2. Model architecture of our applied MLP.



Appendix G. Explanations for Why BRTM is Better than RTM-GH and BRTM-SEP

G.1 Difference in Generated Features

Based on Table 5 in the main text, we first explain how the features generated for the three models are different. In order to isolate the effect of the features derived from bilateral reviews, we kept everything else the same across these three models including having the same number of features, using both profile features and features from item descriptions, how the topic vectors for both the guest and the item are structured, and how the features are constructed from the topics. Therefore, the main difference between these three models is the two highlighted cells in each table (Row1-Column3, Row1-Column4).

Table G.1 for BRTM is the same as Table 5 in the main text. In Table G.2 for RTM-GH where guest reviews and host reviews are merged together into one corpus to derive a single topic space, the topics used to generate the topic vector for $\boldsymbol{A}$ and that for $\boldsymbol{B}$ are the same and they are topics derived from both guest reviews and host reviews. In Column 3, topics related to host reviews but not relevant to guest reviews are also used to represent guest review documents $\boldsymbol{A}$. Similarly, in Column 4, topics related to guest reviews but not relevant to host reviews are also used to represent host review documents $\boldsymbol{B}$. The features constructed from these irrelevant topics will not contribute to the prediction that much. In addition, the topics derived may not be as focused as well since different types of reviews are all in the same corpus. We will further illustrate these two points (i.e., irrelevant features and less focused topics) in Appendix G.2 where we explain using examples from the experimental results.

In order to show the benefit that the bilateral review topics in Table G.1 (BRTM) have over those in Table G.3 (BRTM-SEP) where no shared topics is considered, we use the simulation in Appendix C to show the benefit of using the topic structure with shared and corpus-specific topics. The main advantage is a better generative process that fits the data better leading to better topics.

Table G.1 Features for BRTM used in the Experiments

	Topics for Item Descriptions: ϕ^{D^*} and $\phi^{D\&A}$	**Topics for Guest Reviews:** $\phi^{D\&A}$, ϕ^{A^*}, and $\phi^{A\&B}$	**Topics for Host Reviews:** $\phi^{A\&B}$ and ϕ^{B^*}	**Profile Features**
Topic Vector 1 for guest g_i	Topic Vector for D_{L_i}	Topic Vector for $A_{(i,\cdot)}$	Topic Vector for $B_{(\cdot,i)}$	
Topic Vector 2 for item l_j and its host h_k	Topic Vector for D_j	Topic Vector for $A_{(\cdot,j)}$	Topic Vector for $B_{(k,\cdot)}$	
Feature Vector	f_D	f_A	f_B	$f_{C,G}$ and $f_{C,H}$

Table G.2 Features for RTM-GH used in the Experiments

	Topics for Item Descriptions: ϕ^D	**Topics for Guest Reviews & Host Reviews Combined**	**Topics for Guest Reviews & Host Reviews Combined**	**Profile Features**
Topic Vector 1 for guest g_i	Topic Vector for D_{L_i}	Topic Vector for $A_{(i,\cdot)}$	Topic Vector for $B_{(\cdot,i)}$	
Topic Vector 2 for item l_j and its host h_k	Topic Vector for D_j	Topic Vector for $A_{(\cdot,j)}$	Topic Vector for $B_{(k,\cdot)}$	
Feature Vector	f_D	f_A	f_B	$f_{C,G}$ and $f_{C,H}$

Table G.3 Features for BRTM-SEP used in the Experiments

	Topics for Item Descriptions: ϕ^D	**Topics for Guest Reviews:** ϕ^A	**Topics for Host Reviews:** ϕ^B	**Profile Features**
Topic Vector 1 for guest g_i	Topic Vector for D_{L_i}	Topic Vector for $A_{(i,\cdot)}$	Topic Vector for $B_{(\cdot,i)}$	
Topic Vector 2 for item l_j and its host h_k	Topic Vector for D_j	Topic Vector for $A_{(\cdot,j)}$	Topic Vector for $B_{(k,\cdot)}$	
Feature Vector	f_D	f_A	f_B	$f_{C,G}$ and $f_{C,H}$

G.2 Example Illustration

We use the following examples found in the experimental results to demonstrate the advantages Table G.1 (BRTM) has over Table G.2 (RTM-GH).

1, Compared to RTM-GH, BRTM can generate more focused topics.

We went through the topics discovered under both models, and found that the topics under BRTM are more focused than those under RTM-GH where bilateral reviews are treated as unilateral reviews. Below, we pick one example topic to illustrate. Table G.4(a) shows a corpus-specific host review topic about "fun talking" identified by BRTM-Sample. Among topics learned by RTM-GH, there are two topics that are most relevant to "fun talking", as shown in Table G.4(b). Although these two topics contain content similar to "fun talking", they are also mixed with other perspectives such as "considerate guest" and "highly recommended". Compared to the topics learned by RTM-GH, the topics identified by BRTM-Sample overall are more focused, which can lead to better features used for prediction.

Table G.4. Learned topics about "fun talking".

(a) By BRTM-Sample.

talk	0.1210
fun	0.1114
time	0.0451
great	0.0443
couple	0.0348
lot	0.0330
people	0.0296
interesting	0.0244
good	0.0213
meet	0.0174

(b) By RTM-GH.

guest	0.0442	recommend	0.1579
considerate	0.0222	friendly	0.1422
share	0.0211	host	0.0948
fun	0.0196	highly	0.0614
time	0.0192	talk	0.0569
interesting	0.0180	guest	0.0464
conversation	0.0177	enjoy	0.0400
enjoy	0.0175	respectful	0.0258
delightful	0.0113	person	0.0257
talk	0.0109	conversation	0.0196

2, RTM-GH generated some irrelevant features.

We use one example from the experimental results to illustrate what we mean by irrelevant features. A topic named "follow house rules" is one of the topics learned by RTM-GH where bilateral reviews are mixed together. The top representative words of this topic are displayed in Table G.5.

Table G.5. Representative words of topic "follow house rules" learned by RTM-GH.

"follow house rules"	
rule	0.1834
house	0.1724
guest	0.1000
respect	0.0790
follow	0.0775

Since we use the same topic space to extract topics from both guest and host reviews in RTM-GH, all the learned topics including the topic about "follow house rules" are used to represent both guest review documents and host review documents. This topic mainly applies to host reviews than guest reviews, because guests rarely mention content related to "follow house rules". As illustrated in Table G.6, values of the dimension "follow house rules" in topic vectors for guest review documents vary very little compared to those in topic vectors for host review documents, which indicates that the dimension "follow house rules" barely differentiates guest review documents making the feature derived from this topic irrelevant for the prediction problem.

Table G.6. Descriptive statistics for values of the dimension "follow house rules" in topic vectors of guest and host reviews.

	Median	**Mean**	**S.D.**
Guest Review Documents	0.000338	0.00131	0.00425
Host Reviews Documents	0.001966	0.02283	0.0477

Compared to RTM-GH, our BRTM model not only captures the topic "follow house rules" but also identifies it as a topic specific for host reviews. Since it is a corpus-specific topic for host reviews, this topic is only used to describe host reviews, and not used to represent guest reviews.

Appendix H. Determination of the Number of Topics

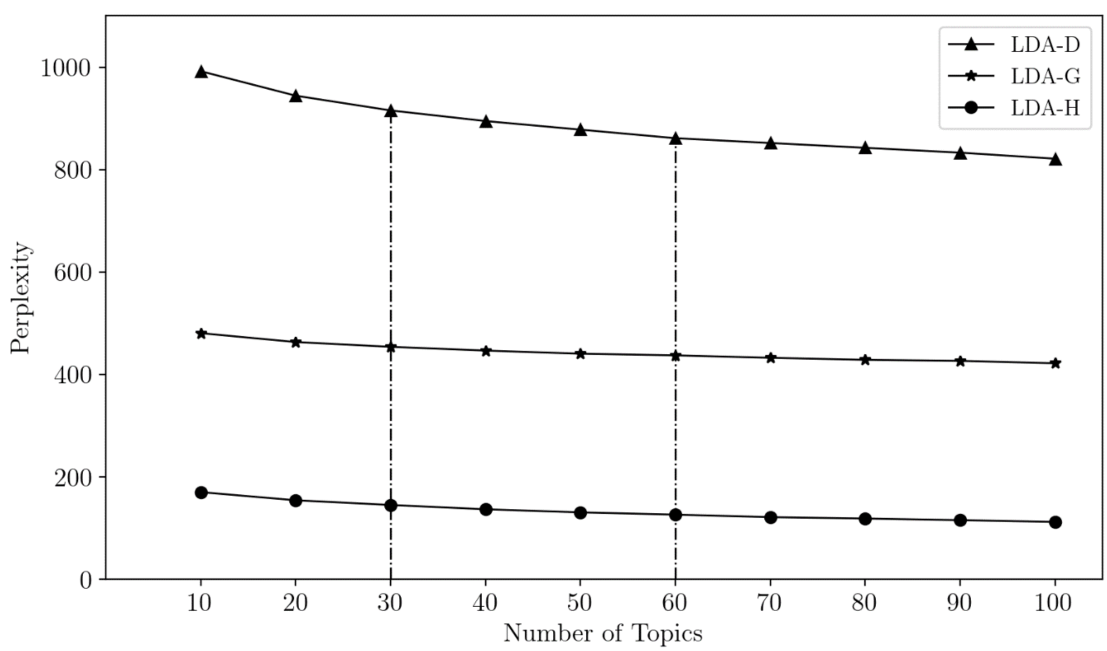
For the models which involve LDA and RTM, we provide the number of topics as a fixed parameter before the learning process starts. Here, we first decide the number of topics for each type of documents, and then estimate the number of shared and corpus-specific topics. Detailed procedures are described as follows.

First of all, we apply LDA-based models on the training set by varying the number of topics from 10 to 100 in steps of 10. We adopt the perplexity (Blei et al. 2003), a widely-used measurement in language modeling, to determine an appropriate number of topics. Based on Equation (I.1), we can compute the perplexity of the validation set, which is held out of the training set, to evaluate the models obtained under different numbers of topics. It describes the degree of uncertainty a topic model has in predicting probabilities of a given text. A lower perplexity score indicates better generalization performance.

$$perp(S_{hold-out}) = \exp\left(-\frac{\sum_{m=1}^{M}\sum_{n=1}^{N_m}\log p(w_{m,n})}{\sum_{m=1}^{M} N_m}\right). \quad \text{(I.1)}$$

As the number of topics gets bigger, the perplexity score goes lower, which indicates a better generalization performance, and tends to converge. Thus, we choose the number of topics where the perplexity score stops improving significantly. Based on the perplexity scores calculated for the different numbers of topics as shown in Figure H.1, we select a range of the number of topics from 30 to 60 (i.e, 30, 40, 50, 60) for each individual type of documents to investigate further. Later on, we will implement our BRTM model under each of the four choices for the number of topics, and pick one which achieves the best performance in prediction on the validation set.

Figure H.1. The perplexity as the number of topics for each type of documents varies.



Before we implement our BRTM model, we need to further determine the number of shared topics under each choice for the number of topics for individual type of documents (i.e. 30, 40, 50, or 60). Every topic learned from LDA-based model corresponds to a word distribution. With these topic-specific distributions, we can examine the similarity between learned topics. Taking topics respectively learned from item description documents (**D**) and guest review documents (**A**) as an example, for each pair of topics from both sides, we can calculate the cosine similarity between them according to Equation (I.2). Since all topic-specific distributions are positive, the range of the cosine similarity is (0, 1).

$$\cos\_sim(\overline{\phi_i^D}, \overline{\phi_j^A}) = \frac{\overline{\phi_i^D} \cdot \overline{\phi_j^A}}{\left\|\overline{\phi_i^D}\right\| \left\|\overline{\phi_j^A}\right\|}. \tag{I.2}$$

A high similarity value means that the pair of topics shares many representative words with similar distributions. An example pair of topics is provided in Table H.1. Both topics are about view of the listing and the cosine similarity between them is 0.75.

Table H.1. An example to illustrate a pair of topics with high similarity.

Item Descriptions		Guest Reviews	
view	0.0958	view	0.1564
roof	0.0322	place	0.0402
deck	0.0254	stay	0.0289
terrace	0.0248	amazing	0.0286
balcony	0.0220	great	0.0180

To estimate the number of shared topics, we first rank all pairs by their cosine similarity in descending order. Then, we remove those pairs dominated by some others. For a given pair <$\overline{\phi_i^D}$, $\overline{\phi_j^A}$>, if there exists another pair that has a higher similarity and shares a topic with the given one, e.g., <$\overline{\phi_i^D}$, $\overline{\phi_{j^*}^A}$> or <$\overline{\phi_{i^*}^D}$, $\overline{\phi_j^A}$>, we define <$\overline{\phi_i^D}$, $\overline{\phi_j^A}$> as a dominated pair and filter it out from the list of candidate pairs. Based on the updated list of pairs, any pair with a similarity greater than 0.5 is regarded as a shared topic. As the number of topics for each individual type of documents varies from 30 to 60 (i.e, 30, 40, 50, 60), the number of shared topics can be correspondingly calculated as summarized in Table H.2. Note that we also considered shared topics between item descriptions and host reviews, and found that there are barely any shared topics between these two types of documents which is consistent with the intuition.

Table H.2. The summary of the number of shared topics.

# of topics for each individual type of documents	# of topics specific for *D*	# of topics shared by *D* and *A*	# of topics specific for *A*	# of topics shared by *A* and *B*	# of topics specific for *B*
30	18	12	12	6	24
40	24	16	15	9	31

50	28	22	18	10	40
60	38	22	29	9	51

Afterward, we implement our BRTM model with the above four groups of parameters and pick one which achieves the best performance in prediction on the validation set. As shown in Table H.3, when varying the number of topics for each individual type of documents from 30 to 60, all three metrics used to evaluate predictive accuracy gradually increase and reach the highest at 60. We pick the group of parameters with the best predictive performance as the final decision of the number of topics. In specific, the number of corpus-specific topics for each corpus is T_{D^*} = 38, T_{A^*} = 29, and T_{B^*} = 51 respectively, and the number of shared topics is set as $T_{D\&A}$ = 22 and $T_{A\&B}$ = 9.

Table H.3. Predictive performances on the validation set with different parameters.

(a) Hit Rate.

Top-N	# of topics for each type of documents			
	30	40	50	60
1	0.175	0.179	0.182	**0.189**
2	0.321	0.320	0.321	**0.343**
3	0.432	0.438	0.440	**0.467**
4	0.534	0.536	0.546	**0.571**
5	0.621	0.619	0.631	**0.662**
6	0.693	0.692	0.709	**0.741**
7	0.754	0.754	0.773	**0.804**
8	0.803	0.800	0.829	**0.856**
9	0.841	0.843	0.871	**0.897**
10	0.875	0.877	0.903	**0.926**

(b) MRR.

Top-N	# of topics for each type of documents			
	30	40	50	60
1	0.175	0.179	0.182	**0.189**
2	0.248	0.250	0.252	**0.266**
3	0.285	0.289	0.291	**0.307**
4	0.311	0.313	0.318	**0.333**
5	0.328	0.330	0.335	**0.352**
6	0.340	0.342	0.348	**0.365**
7	0.349	0.351	0.357	**0.374**
8	0.355	0.357	0.364	**0.380**
9	0.359	0.362	0.369	**0.385**
10	0.363	0.365	0.372	**0.388**

(c) NDCG.

Top-N	# of topics for each type of documents			
	30	**40**	**50**	**60**
1	0.175	0.179	0.182	**0.189**
2	0.267	0.268	0.270	**0.286**
3	0.323	0.327	0.329	**0.348**
4	0.367	0.369	0.375	**0.393**
5	0.400	0.401	0.408	**0.428**
6	0.426	0.427	0.436	**0.456**
7	0.446	0.448	0.457	**0.477**
8	0.462	0.463	0.475	**0.494**
9	0.473	0.476	0.487	**0.506**
10	0.483	0.485	0.497	**0.514**

Appendix I. Comparison of the Predictive Performances

I.1 Additional Predictive Performances of BRTM-Sample

In addition to the improvement of Hit Rate, our proposed BRTM-Sample model also achieves significant improvements in both MRR and NDCG, as illustrated by Tables I.1 and I.2 respectively.

Table I.1. Comparison of MRR between BRTM-Sample and baselines.

Top-N	RAND	RANK-RATING	CF-G	LDA-G	STL	DL-CNN	RTM-G	RTM-GH	BRTM-SEP	BRTM-Sample
1	0.050 (+304.8%)	0.067 (+204.2%)	0.099 (+106.4%)	0.110 (+84.8%)	0.121 (+69.0%)	0.167 (+22.0%)	0.110 (+85.4%)	0.177 (+15.3%)	0.179 (+14.0%)	**0.204**
2	0.075 (+278.0%)	0.103 (+174.7%)	0.138 (+106.0%)	0.158 (+79.3%)	0.177 (+60.6%)	0.235 (+20.8%)	0.162 (+74.8%)	0.245 (+15.9%)	0.248 (+14.4%)	**0.284**
3	0.092 (+257.2%)	0.130 (+151.2%)	0.161 (+103.0%)	0.187 (+74.4%)	0.213 (+53.5%)	0.273 (+19.8%)	0.195 (+67.9%)	0.284 (+15.0%)	0.287 (+13.9%)	**0.327**
4	0.104 (+240.4%)	0.151 (+135.1%)	0.178 (+99.5%)	0.209 (+70.1%)	0.238 (+49.5%)	0.296 (+19.9%)	0.218 (+62.9%)	0.309 (+14.9%)	0.312 (+13.7%)	**0.355**
5	0.114 (+227.2%)	0.168 (+122.7%)	0.191 (+95.5%)	0.225 (+66.3%)	0.257 (+45.6%)	0.312 (+19.7%)	0.235 (+58.9%)	0.326 (+14.7%)	0.330 (+13.3%)	**0.373**
6	0.122 (+216.5%)	0.181 (+113.2%)	0.202 (+91.7%)	0.237 (+63.2%)	0.271 (+42.6%)	0.324 (+19.5%)	0.248 (+55.7%)	0.338 (+14.4%)	0.342 (+13.1%)	**0.387**
7	0.129 (+205.7%)	0.192 (+105.5%)	0.211 (+87.8%)	0.247 (+60.1%)	0.282 (+40.1%)	0.332 (+19.0%)	0.259 (+53.0%)	0.347 (+14.1%)	0.351 (+12.8%)	**0.396**
8	0.135 (+196.2%)	0.202 (+98.9%)	0.218 (+84.1%)	0.255 (+57.1%)	0.291 (+37.9%)	0.338 (+18.6%)	0.267 (+50.4%)	0.353 (+13.6%)	0.357 (+12.4%)	**0.401**
9	0.141 (+187.7%)	0.209 (+94.2%)	0.224 (+80.6%)	0.262 (+54.7%)	0.297 (+36.3%)	0.343 (+18.1%)	0.273 (+48.3%)	0.358 (+13.1%)	0.362 (+12.0%)	**0.405**
10	0.146 (+179.9%)	0.214 (+90.6%)	0.230 (+77.3%)	0.267 (+52.4%)	0.302 (+35.0%)	0.347 (+17.6%)	0.278 (+46.5%)	0.362 (+12.6%)	0.365 (+11.6%)	**0.408**

Note: STL, DL-CNN, RTM-G, RTM-GH, and BRTM-SEP all use the same negative instances as those in BRTM-Sample for fair comparisons.

Table I.2. Comparison of NDCG between BRTM-Sample and baselines.

Top-N	RAND	RANK-RATING	CF-G	LDA-G	STL	DL-CNN	RTM-G	RTM-GH	BRTM-SEP	BRTM-Sample
1	0.050 (+304.8%)	0.067 (+204.2%)	0.099 (+106.4%)	0.110 (+84.8%)	0.121 (+69.0%)	0.167 (+22.0%)	0.110 (+85.4%)	0.177 (+15.3%)	0.179 (+14.0%)	**0.204**
2	0.081 (+273.7%)	0.113 (+170.1%)	0.148 (+105.9%)	0.171 (+78.4%)	0.191 (+59.2%)	0.253 (+20.6%)	0.176 (+73.0%)	0.262 (+16.1%)	0.266 (+14.5%)	**0.305**
3	0.106 (+247.7%)	0.153 (+141.3%)	0.183 (+102.0%)	0.215 (+72.1%)	0.246 (+50.2%)	0.310 (+19.3%)	0.225 (+64.5%)	0.322 (+14.8%)	0.325 (+13.8%)	**0.369**
4	0.128 (+225.9%)	0.189 (+121.1%)	0.212 (+97.1%)	0.251 (+66.4%)	0.288 (+45.0%)	0.350 (+19.5%)	0.265 (+57.8%)	0.365 (+14.6%)	0.368 (+13.5%)	**0.418**
5	0.147 (+207.9%)	0.221 (+105.0%)	0.237 (+91.1%)	0.282 (+60.9%)	0.325 (+39.6%)	0.381 (+19.2%)	0.298 (+52.4%)	0.397 (+14.3%)	0.401 (+13.0%)	**0.454**
6	0.165 (+193.0%)	0.251 (+92.4%)	0.099 (+85.3%)	0.308 (+56.2%)	0.356 (+35.2%)	0.405 (+18.9%)	0.326 (+47.7%)	0.423 (+13.9%)	0.428 (+12.6%)	**0.482**
7	0.181 (+177.2%)	0.276 (+81.7%)	0.148 (+78.9%)	0.332 (+51.3%)	0.382 (+31.5%)	0.425 (+18.1%)	0.350 (+43.6%)	0.443 (+13.3%)	0.448 (+12.0%)	**0.502**

8	0.197 (+162.9%)	0.300 (+72.4%)	0.183 (+72.5%)	0.353 (+46.4%)	0.404 (+27.9%)	0.441 (+17.3%)	0.371 (+39.4%)	0.460 (+12.3%)	0.464 (+11.3%)	**0.517**
9	0.211 (+149.9%)	0.318 (+65.7%)	0.212 (+66.5%)	0.371 (+42.3%)	0.421 (+25.2%)	0.453 (+16.4%)	0.388 (+35.9%)	0.473 (+11.4%)	0.477 (+10.5%)	**0.527**
10	0.225 (+137.6%)	0.334 (+60.2%)	0.237 (+60.7%)	0.387 (+38.3%)	0.434 (+23.1%)	0.464 (+15.3%)	0.403 (+32.7%)	0.484 (+10.4%)	0.487 (+9.68%)	**0.535**

Note: STL, DL-CNN, RTM-G, RTM-GH, and BRTM-SEP all use the same negative instances as those in BRTM-Sample for fair comparisons.

I.2 Predictive Performances of BRTM-Rank

BRTM-Rank also performs the best among all the alternatives. The comparisons of Hit Rate, MRR and NDCG are displayed respectively in Tables I.3, I.4, and I.5.

Table I.3. Comparison of Hit Rate between BRTM-Rank and representative baselines.

Top-N	**RAND**	**RANK-RATING**	**CF-G**	**LDA-G**	**STL**	**DL-CNN**	**RTM-G**	**RTM-GH**	**BRTM-SEP**	**BRTM-Sample Random**
1	0.050 (+267.5%)	0.067 (+176.2%)	0.099 (+87.4%)	0.110 (+67.8%)	0.121 (+53.5%)	0.167 (+10.8%)	0.114 (+63.1%)	0.167 (+11.0%)	0.170 (+9.04%)	**0.185**
2	0.100 (+230.5%)	0.140 (+136.2%)	0.177 (+86.5%)	0.206 (+59.9%)	0.233 (+41.7%)	0.303 (+8.89%)	0.220 (+49.9%)	0.302 (+8.95%)	0.307 (+7.46%)	**0.330**
3	0.149 (+201.3%)	0.220 (+104.2%)	0.247 (+82.4%)	0.294 (+53.0%)	0.342 (+31.5%)	0.417 (+7.91%)	0.315 (+42.6%)	0.421 (+6.90%)	0.428 (+5.00%)	**0.450**
4	0.200 (+174.8%)	0.304 (+81.3%)	0.314 (+75.0%)	0.379 (+45.3%)	0.440 (+25.1%)	0.510 (+7.95%)	0.404 (+36.1%)	0.527 (+4.43%)	0.534 (+3.04%)	**0.550**
5	0.250 (+156.7%)	0.387 (+65.6%)	0.380 (+68.7%)	0.458 (+39.9%)	0.535 (+19.8%)	0.590 (+8.63%)	0.487 (+31.6%)	0.620 (+3.34%)	0.627 (+2.21%)	**0.641**
6	0.298 (+141.5%)	0.469 (+53.3%)	0.444 (+62.1%)	0.533 (+35.0%)	0.624 (+15.4%)	0.659 (+9.12%)	0.566 (+27.1%)	0.702 (+2.51%)	0.707 (+1.77%)	**0.719**
7	0.348 (+125.4%)	0.547 (+43.4%)	0.506 (+55.1%)	0.603 (+29.9%)	0.700 (+12.0%)	0.719 (+9.06%)	0.637 (+23.1%)	0.775 (+1.22%)	0.773 (+1.39%)	**0.784**
8	0.397 (+111.8%)	0.621 (+35.3%)	0.565 (+48.7%)	0.670 (+25.4%)	0.770 (+9.09%)	0.768 (+9.43%)	0.703 (+19.5%)	0.826 (+1.71%)	0.831 (+1.09%)	**0.840**
9	0.445 (+98.5%)	0.683 (+29.3%)	0.622 (+41.9%)	0.729 (+21.1%)	0.827 (+6.72%)	0.809 (+9.08%)	0.764 (+15.6%)	0.874 (+1.03%)	0.877 (+0.62%)	**0.883**
10	0.493 (+86.2%)	0.736 (+24.8%)	0.677 (+35.5%)	0.784 (+17.1%)	0.873 (+5.20%)	0.845 (+8.64%)	0.814 (+12.8%)	0.911 (+0.81%)	0.914 (+0.48%)	**0.918**

Note: RTM-G, RTM-GH, and BRTM-SEP all use the BPR loss function as that in BRTM-Rank for fair comparisons.

Table I.4. Comparison of MRR between BRTM-Rank and representative baselines.

Top-N	**RAND**	**RANK-RATING**	**CF-G**	**LDA-G**	**STL**	**DL-CNN**	**RTM-G**	**RTM-GH**	**BRTM-SEP**	**BRTM-Sample Random**
1	0.050 (+267.5%)	0.067 (+176.2%)	0.099 (+87.4%)	0.110 (+67.8%)	0.121 (+53.5%)	0.167 (+10.8%)	0.114 (+63.1%)	0.167 (+11.0%)	0.170 (+9.04%)	**0.185**
2	0.075 (+242.9%)	0.103 (+149.2%)	0.138 (+86.8%)	0.158 (+62.7%)	0.177 (+45.7%)	0.235 (+9.57%)	0.167 (+54.4%)	0.235 (+9.67%)	0.238 (+8.02%)	**0.257**
3	0.092 (+224.8%)	0.130 (+128.4%)	0.161 (+84.7%)	0.187 (+58.6%)	0.213 (+39.6%)	0.273 (+8.98%)	0.199 (+49.8%)	0.274 (+8.51%)	0.279 (+6.68%)	**0.297**

4	0.104 (+209.2%)	0.151 (+113.6%)	0.178 (+81.2%)	0.209 (+54.6%)	0.238 (+35.8%)	0.296 (+8.91%)	0.221 (+46.1%)	0.301 (+7.29%)	0.305 (+5.68%)	**0.323**
5	0.114 (+198.3%)	0.168 (+103.1%)	0.191 (+78.3%)	0.225 (+51.7%)	0.257 (+32.8%)	0.312 (+9.12%)	0.237 (+43.5%)	0.319 (+6.70%)	0.324 (+5.20%)	**0.341**
6	0.122 (+189.4%)	0.181 (+95.0%)	0.202 (+75.3%)	0.237 (+49.3%)	0.271 (+30.4%)	0.324 (+9.27%)	0.250 (+41.2%)	0.333 (+6.27%)	0.337 (+4.93%)	**0.354**
7	0.129 (+180.6%)	0.192 (+88.6%)	0.211 (+72.4%)	0.247 (+46.9%)	0.282 (+28.6%)	0.332 (+9.24%)	0.261 (+39.3%)	0.343 (+5.74%)	0.347 (+4.73%)	**0.363**
8	0.135 (+173.1%)	0.202 (+83.4%)	0.218 (+69.7%)	0.255 (+44.9%)	0.291 (+27.1%)	0.338 (+9.35%)	0.269 (+37.6%)	0.351 (+5.46%)	0.354 (+4.57%)	**0.370**
9	0.141 (+166.1%)	0.209 (+79.6%)	0.224 (+67.0%)	0.262 (+43.1%)	0.297 (+26.0%)	0.343 (+9.25%)	0.276 (+36.0%)	0.356 (+5.19%)	0.359 (+4.39%)	**0.375**
10	0.146 (+159.7%)	0.214 (+76.8%)	0.230 (+64.5%)	0.267 (+41.4%)	0.302 (+25.3%)	0.347 (+9.15%)	0.281 (+34.8%)	0.360 (+5.09%)	0.363 (+4.32%)	**0.378**

Note: RTM-G, RTM-GH, and BRTM-SEP all use the BPR loss function as that in BRTM-Rank for fair comparisons.

Table I.5. Comparison of NDCG between BRTM-Rank and representative baselines.

Top-N	**RAND**	**RANK-RATING**	**CF-G**	**LDA-G**	**STL**	**DL-CNN**	**RTM-G**	**RTM-GH**	**BRTM-SEP**	**BRTM-Sample Pop**
1	0.050 (+267.5%)	0.067 (+176.2%)	0.099 (+87.4%)	0.110 (+67.8%)	0.121 (+53.5%)	0.167 (+10.8%)	0.114 (+63.1%)	0.167 (+11.0%)	0.170 (+9.04%)	**0.185**
2	0.081 (+239.0%)	0.113 (+145.0%)	0.148 (+86.7%)	0.171 (+61.8%)	0.191 (+44.4%)	0.253 (+9.35%)	0.181 (+52.9%)	0.252 (+9.44%)	0.256 (+7.85%)	**0.276**
3	0.106 (+216.5%)	0.153 (+119.6%)	0.183 (+83.9%)	0.215 (+56.7%)	0.246 (+36.7%)	0.310 (+8.61%)	0.228 (+47.3%)	0.311 (+7.96%)	0.317 (+6.11%)	**0.336**
4	0.128 (+196.1%)	0.189 (+100.8%)	0.212 (+79.0%)	0.251 (+51.1%)	0.288 (+31.8%)	0.350 (+8.56%)	0.267 (+42.4%)	0.357 (+6.26%)	0.362 (+4.73%)	**0.380**
5	0.147 (+181.4%)	0.221 (+87.4%)	0.237 (+74.6%)	0.282 (+47.1%)	0.325 (+27.6%)	0.381 (+8.91%)	0.299 (+38.8%)	0.393 (+5.43%)	0.398 (+4.07%)	**0.415**
6	0.165 (+169.0%)	0.251 (+76.6%)	0.260 (+70.1%)	0.308 (+43.5%)	0.356 (+24.2%)	0.405 (+9.18%)	0.327 (+35.4%)	0.422 (+4.79%)	0.427 (+3.69%)	**0.443**
7	0.181 (+156.2%)	0.276 (+67.9%)	0.281 (+65.3%)	0.332 (+39.8%)	0.382 (+21.6%)	0.425 (+9.14%)	0.350 (+32.4%)	0.447 (+3.92%)	0.449 (+3.38%)	**0.464**
8	0.197 (+145.1%)	0.300 (+60.7%)	0.300 (+60.9%)	0.353 (+36.5%)	0.404 (+19.3%)	0.441 (+9.34%)	0.371 (+29.8%)	0.466 (+3.47%)	0.467 (+3.13%)	**0.482**
9	0.211 (+134.4%)	0.318 (+55.4%)	0.317 (+56.2%)	0.371 (+33.5%)	0.421 (+17.5%)	0.453 (+9.16%)	0.390 (+27.0%)	0.480 (+2.98%)	0.481 (+2.81%)	**0.495**
10	0.225 (+124.4%)	0.334 (+51.3%)	0.333 (+51.7%)	0.387 (+30.6%)	0.434 (+16.2%)	0.464 (+8.93%)	0.404 (+25.0%)	0.491 (+2.82%)	0.492 (+2.69%)	**0.505**

Note: RTM-G, RTM-GH, and BRTM-SEP all use the BPR loss function as that in BRTM-Rank for fair comparisons.

In our experiments, we actually implemented two deep learning models (DL-MLP and DL-CNN) as baselines for comparison. Because DL-MLP performs worse than DL-CNN as shown in Table I.6, we only included the results of DL-CNN in the main comparison tables.

Table I.6. Comparison of Hit Rate between DL-MLP and DL-CNN.

Top-N	DL-MLP	DL-CNN
1	0.157	0.167
2	0.290	0.303
3	0.402	0.417
4	0.499	0.510
5	0.582	0.590
6	0.652	0.659
7	0.710	0.719
8	0.761	0.768
9	0.805	0.809
10	0.840	0.845

I.3 Results for setting the number of topics as 30

In addition, we lower the number of topics for each individual type of documents from 60 to 30 to further check the robustness of our method. As illustrated by the comparisons in Table I.7, the conclusion that our proposed BRTM model outperforms other alternatives still remains.

Table I.7. Comparison of Hit Rate when # of topics for each individual type of documents is 30.

Top-N	RAND	RANK-RATING	CF-G	LDA-G	STL	DL-CNN	RTM-G	RTM-GH	BRTM-SEP	BRTM-Sample
1	0.050 (+268.0%)	0.067 (+176.6%)	0.099 (+87.6%)	0.113 (+63.9%)	0.121 (+10.9%)	0.167 (+18.1%)	0.119 (+55.6%)	0.171 (+8.57%)	0.168 (+10.2%)	**0.185**
2	0.100 (+233.3%)	0.140 (+138.2%)	0.177 (+88.1%)	0.215 (+54.6%)	0.233 (+9.79%)	0.303 (+14.6%)	0.230 (+44.7%)	0.305 (+8.98%)	0.302 (+9.86%)	**0.332**
3	0.149 (+204.3%)	0.220 (+106.2%)	0.247 (+84.2%)	0.308 (+47.3%)	0.342 (+8.97%)	0.417 (+12.9%)	0.329 (+38.1%)	0.417 (+8.84%)	0.417 (+9.02%)	**0.454**
4	0.200 (+177.3%)	0.304 (+82.9%)	0.314 (+76.6%)	0.393 (+41.3%)	0.440 (+8.94%)	0.510 (+11.2%)	0.419 (+32.6%)	0.515 (+7.87%)	0.513 (+8.33%)	**0.555**
5	0.250 (+158.6%)	0.387 (+66.8%)	0.380 (+69.9%)	0.472 (+36.6%)	0.535 (+9.42%)	0.590 (+10.9%)	0.503 (+28.3%)	0.600 (+7.60%)	0.597 (+8.02%)	**0.645**
6	0.298 (+142.0%)	0.469 (+53.6%)	0.444 (+62.4%)	0.547 (+31.8%)	0.624 (+15.6%)	0.659 (+9.36%)	0.579 (+24.4%)	0.672 (+7.30%)	0.669 (+7.76%)	**0.721**
7	0.348 (+125.9%)	0.547 (+43.7%)	0.506 (+55.4%)	0.616 (+27.5%)	0.700 (+12.3%)	0.719 (+9.29%)	0.650 (+21.0%)	0.734 (+7.08%)	0.732 (+7.32%)	**0.786**
8	0.397 (+111.1%)	0.621 (+34.9%)	0.565 (+48.2%)	0.681 (+23.1%)	0.770 (+8.73%)	0.768 (+9.07%)	0.714 (+17.3%)	0.789 (+6.21%)	0.787 (+6.41%)	**0.838**
9	0.445 (+97.8%)	0.683 (+28.9%)	0.622 (+41.5%)	0.740 (+18.9%)	0.827 (+6.36%)	0.809 (+8.70%)	0.771 (+14.1%)	0.836 (+5.23%)	0.834 (+5.53%)	**0.880**
10	0.493 (+85.2%)	0.736 (+24.1%)	0.677 (+34.8%)	0.795 (+14.9%)	0.873 (+4.63%)	0.845 (+8.06%)	0.821 (+11.3%)	0.875 (+4.39%)	0.873 (+4.56%)	**0.913**

Note: STL, DL-CNN, RTM-G, RTM-GH, and BRTM-SEP all use the same negative instances as those in BRTM-Sample for fair comparisons.

Appendix J. Details of Topics Investigation

To measure the coherence of the 149 topics discovered from all the three types of documents, we recruited eight master workers from Amazon MTurk to perform the word intrusion task designed by Chang et al. (2009). For each topic, each worker is presented with six randomly ordered words including the top-5 key words from the topic and one random word which is the intruder word. The task of the workers is to pick the intruder word. Following the procedure proposed in Chang et al. (2009), an intruder word is randomly selected from a pool of words with low probability in the current topic but high probability in some other topic to ensure that the intruder is not picked solely because it is a rare word. If the topic is more coherent, it should be easier for the worker to identify the intruder word. The coherence score of the topic can be measured by the percentage of workers picking the correct intruder word. The results show that 80% of the topics have precision over 0.375 (i.e., no fewer than 3 people have chosen the true intrusion word), which is a good coherence value. If one word is randomly chosen from all the 6 words, only 1.33 people will choose the true intrusion word.

For each discovered topic, we name the topic based on its top representative words. Among the words whose probability in the topic-word distribution is greater than 0.005, we pick the top five to serve as the representative words for each topic. With the representative words, we conduct a word intrusion task to evaluate topic coherence. The majority of the topics has decent coherence. From Table J.1 to Table J.5, we summarize different types of topics by displaying the topic name, the representative words, the percentage of documents covered (using 0.01 as the cutoff), and the coherence score from the word intrusion task for each discovered topic. For better illustration, we represent each topic with a word cloud in Figures J.1 - J.5. Each word cloud shows the top twenty representative words in each topic, and the size of the word indicates the representativeness of the word for the topic.

Table J.1. Summary of corpus-specific topics from listing description documents *D*.

#	Topic Name	Representative Words	% of Documents *D*	Coherence
1	"bar"	bar, space, music, great, food	20.2%	0.875
2	"trash policy"	leave, trash, garbage, recycle, bin	21.3%	0.75
3	"large area"	large, area, floor, design, view	23.1%	0.875
4	"express ride"	minute, express, neighborhood, ride, museum	24.9%	0.5
5	"share room"	guest, room, share, work, bathroom	34.3%	0.75
6	"mountain resort"	pool, mountain, resort, ski, swim	12.0%	0.875
7	"downtown"	downtown, house, walk, guest, dog	25.6%	0.375
8	"street direction"	street, north, minute, south, west	19.4%	0.625
9	"city apartment"	apartment, city, locate, bed, meter	17.8%	0.5

10	"hot tub"	water, hot, guest, tub, property	11.6%	0.5
11	"private room"	room, private, bathroom, guest, kitchen	61.1%	0.75
12	"walking minutes to subway"	apartment, subway, building, walk, minute	42.1%	0.625
13	"artist neighborhood"	artist, neighborhood, loft, con, space	15.9%	0.25
14	"check-out policy"	time, leave, free, guest, room	19.9%	0.125
15	"bike"	bike, street, walk, park, private	26.1%	0.5
16	"room & bed"	room, bed, free, minute, machine	26.8%	0.375
17	"neighborhood & backyard"	neighborhood, backyard, party, garden, duplex	33.5%	0.5
18	"airport transportation"	station, airport, bus, minute, taxi	39.1%	0.75
19	"garden"	house, garden, villa, room, guest	11.1%	0.875
20	"museum"	museum, city, metro, street, plaza	17.3%	0.75
21	"house"	house, room, guest, time, share	22.4%	0.875
22	"towel & iron"	towel, provide, iron, guest, check	40.3%	0.5
23	"property damage"	property, guest, damage, owner, rental	19.7%	0.625
24	"walking minutes to station"	flat, walk, station, minute, street	18.9%	0.5
25	"location of the studio"	studio, block, best, heart, bar	34.9%	0.125
26	"check-in policy"	check, guest, stay, time, fee	39.4%	0.375
27	"bedroom suite"	bedroom, house, suite, floor, deck	30.8%	0.75
28	"local food"	hidden, food, hide, local, website	15.2%	0.125
29	"city centre & free parking"	walk, city, centre, free, park	17.9%	0.375
30	"driving minutes"	minute, drive, guest, private, bay	19.1%	0.25
31	"miles to park and downtown"	mile, park, downtown, center, street	18.1%	0.75
32	"close to park"	park, garden, prospect, minute, walk	24.9%	0.375
33	"train"	train, block, room, kitchen, bedroom	52.2%	0.5
34	"bed & breakfast"	bed, room, breakfast, enjoy, local	13.8%	0.375
35	"door key"	door, minute, key, walk, leave	32.7%	0.625
36	"loft"	building, loft, bedroom, modern, locate	43.0%	0.625
37	"park & museum"	park, central, museum, side, block	37.7%	0.5
38	"health-related"	guest, organic, yoga, private, garden	16.3%	0.375

Table J.2. Summary of corpus-specific topics from guest review documents *A*.

#	Topic Name	Representative Words	% of Documents *A*	Coherence
1	"beautiful & wonderful stay"	beautiful, wonderful, stay, lovely, feel	33.7%	0.375
2	"comfortable"	stay, time, comfortable, bed, super	29.1%	0.25
3	"cancellation before arrival"	arrival, reservation, day, cancel, posting	26.8%	0.25

4	"key location"	apartment, great, key, stay, location	33.0%	0.5
5	"clean place"	location, room, place, clean, stay	30.0%	0.875
6	"breakfast"	breakfast, stay, great, host, room	23.3%	0.875
7	"great stay"	stay, great, location, definitely, host	54.7%	0.5
8	"nice & good"	nice, good, subway, stay, place	48.9%	0.5
9	"noise"	noise, location, great, noisy, night	33.4%	0.5
10	"restaurant"	location, stay, great, apartment, restaurant	26.2%	0.875
11	"station"	apartment, station, flat, stay, clean	37.0%	0.625
12	"lovely stay"	apartment, stay, great, recommend, lovely	36.6%	0.5
13	"clean subway"	clean, subway, apartment, place, stay	47.1%	0.25
14	"nice place"	place, nice, stay, great, clean	49.8%	0.875
15	"great location"	great, location, stay, place, host	54.7%	0.625
16	"great stay in Prospect Park area"	park, stay, great, prospect, slope	15.0%	0.375
17	"nice room"	room, house, stay, nice, place	34.8%	0.75
18	"perfect place"	place, stay, perfect, beautiful, super	29.3%	0.375
19	"clean room"	room, bathroom, stay, great, clean	44.3%	0.625
20	"bar & restaurant"	great, stay, bar, cool, restaurant	37.2%	0.625
21	"great location"	stay, apartment, great, location, well	39.5%	0.25
22	"central park location"	park, central, apartment, stay, location	27.7%	0.875
23	"rooftop view"	view, apartment, great, place, rooftop	21.3%	0.75
24	"airport"	airport, stay, place, clean, host	28.9%	0.375
25	"great place"	great, place, check, clean, host	60.7%	0.375
26	"loft"	loft, stay, space, place, great	12.5%	0.75
27	"studio location"	studio, location, great, stay, place	23.4%	0.625
28	"dog-related"	dog, stay, place, great, time	13.8%	0
29	"great family"	apartment, family, great, stay, well	31.7%	0.5

Table J.3. Summary of corpus-specific topics from host review documents *B*.

#	Topic Name	Representative Words	% of Documents *B*	Coherence
1	"happy experience"	happy, guest, host, great, happily	37.3%	0.875
2	"family & friend"	family, friend, group, guest, people	32.2%	0.875
3	"leave it immaculate"	condition, leave, apartment, immaculate, host	32.7%	0.375
4	"independent & quiet"	pleasant, recommend, independent, quiet, host	26.1%	0.625
5	"communicate well about plan"	well, communicate, arrival, plan, stay	35.7%	0.375
6	"polite & friendly"	polite, guest, friendly, community, respectful	32.6%	0.625
7	"good to deal with"	guest, deal, good, glad, flat	28.7%	0.25
8	"check-out"	check, guest, problem, leave, host	32.5%	0.375

9	"in good order"	friend, order, good, time, people	27.9%	0.625
10	"considerate & responsive"	considerate, clean, guest, responsive, thoughtful	28.4%	0.5
11	"awesome & amazing guest"	guest, awesome, great, cool, amazing	33.0%	0.5
12	"pleasure to host"	host, pleasure, absolute, great, recommend	56.7%	0.625
13	"definitely recommend"	definitely, host, recommend, friend, stay	35.6%	0.625
14	"easy to communicate"	easy, communicate, host, recommend, leave	52.9%	0.375
15	"follow house rules"	house, rule, respect, follow, clean	32.5%	0.375
16	"duration of stay"	stay, night, week, day, time	41.4%	1
17	"leave it spotless"	apartment, leave, guest, spotless, great	48.4%	0.625
18	"lovely couple"	lovely, guest, husband, warm, couple	30.7%	0.5
19	"enjoy guest visit"	enjoy, stay, chat, guest, visit	34.4%	0.625
20	"friendly"	friendly, recommend, highly, guest, clean	48.6%	0.625
21	"excellent guest"	excellent, guest, rent, communication, care	27.4%	0.5
22	"good communicator"	super, sweet, communicator, respectful, great	38.9%	0.875
23	"glad to host"	gladly, time, guest, host, real	16.6%	0.125
24	"delightful"	wonderful, guest, delightful, delight, time	44.6%	0.875
25	"fun talking"	talk, fun, time, great, couple	33.4%	0.75
26	"amazing people"	guest, host, amazing, extremely, lucky	28.6%	0.5
27	"clean & quiet"	guest, barely, clean, best, quiet	22.3%	0.25
28	"have guest anytime"	anytime, guest, great, clean, leave	45.2%	0.375
29	"enjoyable conversation"	host, conversation, enjoy, speak, work	22.3%	0.5
30	"communicative"	communicative, time, great, guest, organize	35.8%	0.875
31	"check-out"	house, apartment, time, people, leave	13.0%	0.875
32	"treatment"	treat, stay, reservation, host, loft	17.0%	0.5
33	"great guest"	great, guest, recommend, leave, good	62.8%	0.375
34	"booking"	book, stay, open, receive, guest	17.9%	0.375
35	"short stay"	stay, short, guest, perfectly, cat	17.2%	0
36	"kind"	kind, care, courteous, good, great	35.9%	0.5
37	"considerate & friendly"	person, guest, considerate, friendly, share	26.5%	0.375
38	"leave it in good shape"	place, leave, good, guest, clean	45.8%	0.375
39	"neat"	neat, guest, host, neighbor, clean	18.8%	0.125
40	"(no) hesitation"	hesitate, host, hesitation, state, fit	16.3%	0
41	"tidy"	tidy, room, leave, friendly, guest	38.8%	0.875
42	"happy communication"	choose, stay, happy, communication, recommendation	16.1%	0.875
43	"visit city"	city, visit, time, day, traveler	22.6%	1
44	"recommend to future hosts"	future, recommend, guest, host, care	29.4%	0.25
45	"responsible"	space, responsible, host, respectful, guest	22.6%	0.75

46	"leave it in perfect condition"	perfect, leave, guest, condition, studio	34.0%	0.125
47	"nice & clean"	nice, clean, guest, people, person	57.8%	0.875
48	"easy & flexible"	apt, guest, easy, leave, flexible	26.9%	0.125
49	"respectful"	respectful, clean, guest, time, great	58.4%	0.75
50	"absolutely recommend"	absolutely, guest, recommend, stay, cross	21.6%	0
51	"quiet"	quiet, host, respectful, friendly, guest	32.7%	1

Table J.4. Summary of shared topics from both listing description and guest review documents.

#	Topic Name	Representative Words	% of Documents *D*	% of Documents *A*	Coherence
1	"clean & nice"	clean, nice, place, stay, close	38.1%	50.7%	0.75
2	"island & ferry"	island, ferry, place, stay, good	17.1%	9.7%	0.625
3	"unique & great place"	place, space, experience, unique, great	31.7%	21.5%	0.875
4	"block & neighborhood"	block, apartment, neighborhood, street, great	45.7%	30.7%	0.75
5	"time square"	square, time, apartment, location, walk	39.3%	23.7%	0
6	"condo units"	unit, condo, guest, bedroom, leave	22.6%	20.5%	0.125
7	"seaside"	beach, ocean, pool, walk, car	13.7%	11.0%	0.625
8	"cat friendly"	cat, room, host, friendly, sweet	17.5%	11.4%	0.25
9	"room situation"	bed, size, bedroom, kitchen, living	53.3%	34.4%	1
10	"great location"	apt, location, great, expect, clean	27.8%	25.0%	0.625
11	"doors & windows"	room, door, window, work, day	35.2%	32.7%	0.5
12	"stairs & floors"	apartment, stair, floor, building, good	23.7%	33.9%	0.5
13	"east/west village"	village, east, west, apartment, restaurant	29.4%	15.1%	0.625
14	"view"	view, terrace, city, bridge, restaurant	31.8%	15.0%	0.25
15	"town"	town, walk, bus, flat, center	20.1%	23.3%	0.5
16	"well located"	apartment, locate, stay, well, perfect	55.5%	54.6%	0.75
17	"subway"	subway, minute, walk, time, safe	38.7%	41.5%	0.875
18	"cottage & cabin"	cottage, cabin, mile, property, house	11.5%	11.5%	0.875
19	"brownstone"	floor, brownstone, subway, street, height	36.7%	14.2%	0.625
20	"breakfast"	guest, breakfast, host, room, access	26.9%	18.9%	0.25
21	"great place"	place, help, stay, room, great	30.3%	32.5%	0.375
22	"garden"	garden, stay, fridge, fruit, lovely	18.6%	20.3%	0.25

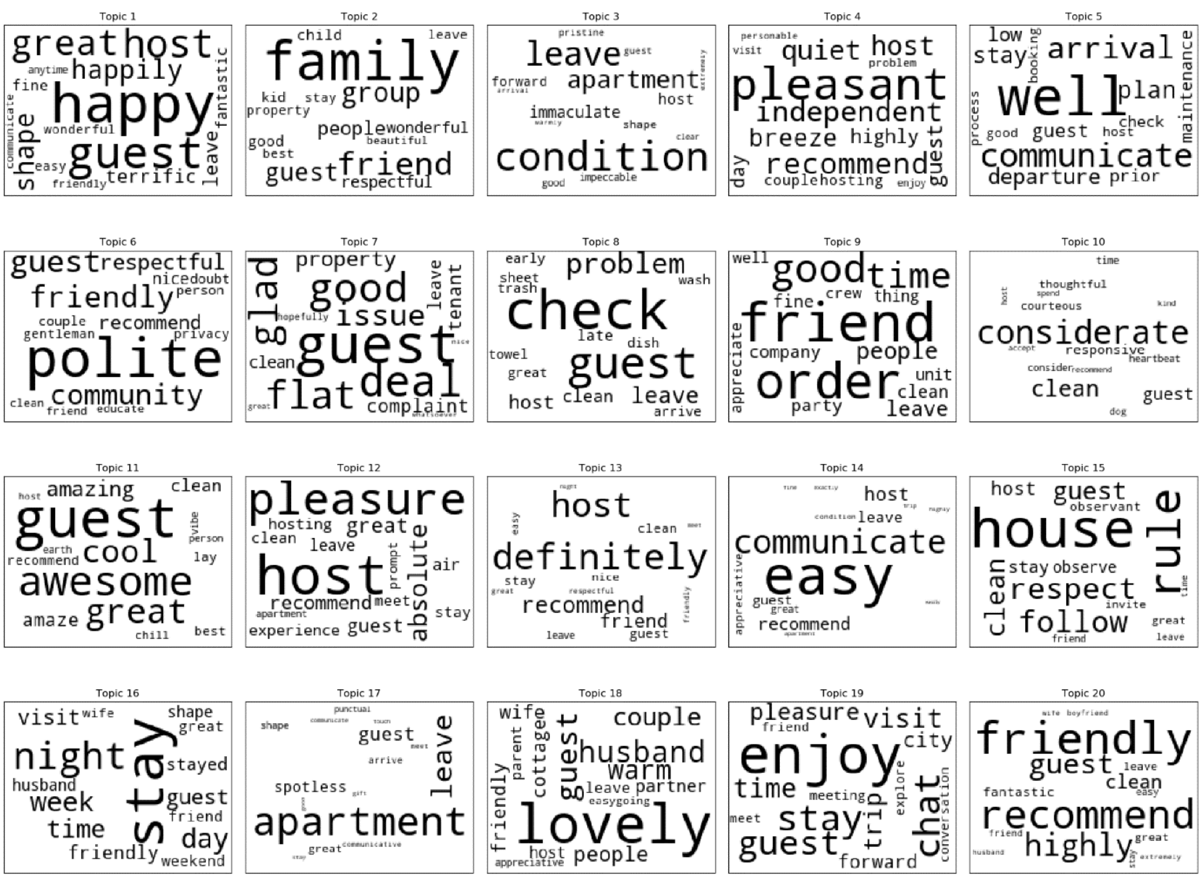
Table J.5. Summary of shared topics from both guest review and host review documents.

#	Topic Name	Representative Words	% of Documents *A*	% of Documents *B*	Coherence
1	"highly recommend"	highly, recommend, host, nice, recommended	39.5%	39.5%	0.125
2	"check-in & out"	meet, stay, leave, time, key	46.6%	32.7%	0.375
3	"smooth communication"	communication, great, smooth, good, experience	28.2%	39.3%	0.75
4	"helpful & comfortable"	host, help, helpful, stay, comfortable	46.8%	17.3%	0.625
5	"great experience"	host, experience, great, feel, friendly	36.3%	16.0%	0.625
6	"stay time"	house, morning, stay, night, day	26.2%	17.5%	0.875
7	"neighborhood & family"	host, neighborhood, wife, time, great	36.6%	12.6%	0
8	"great group"	group, great, work, space, accommodate	33.6%	21.8%	0.625
9	"super clean"	clean, room, host, super, place	38.7%	15.8%	0.25

Figure J.1. The word cloud of each topic learned from listing description documents.

Figure J.2. The word cloud of each topic learned from guest review documents.

Figure J.3. The word cloud of each topic learned from host review documents.




Topic 21
excellent apartment leave people condition apt timely rent communication guest friend care provide highly good
Topic 22
stay couple respectful friend meet sweet girl totally kind nice guest super great clean communicator
Topic 23
great guest real easy friendly gladly coordinate future considerate time provide travel pleasure host communication
Topic 24
meet delight return thoughtful time friend visit husband wonderful guest enjoy appreciative host delightful fabulous
Topic 25
interesting couple smile talk conversation spend people fun lot stay good time great experience meet
Topic 26
amazing people best simply warm felt guest dream energy fantastic forward host lucky pleasure extremely
Topic 27
respectful personality funny barely great quiet thing fun smart stay guest notice best clean
Topic 28
communication anytime stay friend clean fantastic guest pleasure leave apartment great
Topic 29
gracious conversation recommend life stay enjoy well time speak host travel respectful work talk schedule
Topic 30
time great communicative organize guest
Topic 31
leave day respect quiet house apartment neighbor pleasure confidence time well good stay people easy
Topic 32
reservation process treat well leave recommend departure host importantly friend stay loft place clean guest
Topic 33
friend great people good communication cheer guest time wife recommend amazing place leave shape
Topic 34
colleague stay renter loft guest property accept day book condo open receive great respectful work
Topic 35
perfectly cat serve visit stay guest tidy term leave follow meet short return surroundings instruction
Topic 36
great guest recommend apartment kind time courteous host opportunity good stay person care gracious appreciate
Topic 37
person social friendly good share positive thoughtful great guest intelligent pleasure interesting conscientious considerate
Topic 38
great place recommend guest clean easygoing communicable meet shape good leave stay friendly care
Topic 39
clean neighbor guest friendly great wonderful leave host clear good prompt neat pin husband find
Topic 40
thoughtful time trust pleasure host addition correspondence fit worry meet state hesitate hesitation sufficient forward
Topic 41
find time room guest stay leave friend host tidy great enjoy good friendly appreciate clean
Topic 42
respect high humor happy cleanliness recommendation choose host guest stay great sense place communication person
Topic 43
traveler couple community spend visit find explore gift day city share arrive travel time appreciate
Topic 44
fantastic host wonderful communicative guest thoughtful ideal reliable warm future great recommend good highly care
Topic 45
guest warm shoe care mindful responsible recommend quiet highly space clean friend host respectful great
Topic 46
easy pleasure time person leave meet apartment highly perfect communication spotless guest studio great condition
Topic 47
clean friend guest good meet couple quiet girl nice person friendly people leave great recommend
Topic 48
key clean apt leave easy work ideal flexible coordinate stay guest plant communication spotless
Topic 49
guest time clean great extremely fantastic boyfriend respectful
Topic 50
sociable time recommend day guest stay cross path schedule absolutely respectful busy friendly manner
Topic 51
friendly pleasure recommend quiet guest great talk friend clean host respectful

Figure J.4. The word cloud of each topic learned from both listing description and guest review documents.

Figure J.5. The word cloud of each topic learned from both guest review and host review documents.

Appendix K. Example Topics Illustrating the Topic Structure

Topics learned by our proposed model align well with the motivation of introducing shared and corpus-specific topics. After examining the representative words of both shared and corpus-specific topics discovered, we are able to identify shared topics that reflect common talking points embedded in both listing descriptions provided by hosts and reviews left by guests. As illustrated by the example shared topic in Table K.1, the first two topics are discovered from listing descriptions and guest reviews separately by running LDA, which contain similar topics about the view of the listing. Under our BRTM framework with shared and corpus-specific topics, the two similar topics are automatically consolidated into one shared topic which is the third topic in Table K.1.

Table K.1. An example shared topic about "view".

Item Descriptions		Guest Reviews		Shared	
view	0.0958	view	0.1564	view	0.0800
roof	0.0322	place	0.0402	terrace	0.0286
deck	0.0254	stay	0.0289	city	0.0253
terrace	0.0248	amazing	0.0286	bridge	0.0238
balcony	0.0220	great	0.0180	restaurant	0.0213

In Table K.2, we also provide some examples showing two shared topics discovered from guest reviews and host reviews under BRTM. Both guests and hosts discuss the smooth communication between each other and highly recommend each other to future hosts or guests.

Table K.2. Examples of shared topics learned from bilateral reviews.

"smooth communication"		"highly recommended"	
communication	0.3017	highly	0.2979
great	0.2895	recommend	0.2301
smooth	0.0407	host	0.0842
good	0.0374	nice	0.0631
experience	0.0326	recommended	0.0534

Meanwhile, topics which are specific for different types of text inputs can also be recognized by our proposed model. Table K.3 presents several corpus-specific topics discovered in our dataset. In Table K.3(a), topics specific for listing description are associated with topics which are unlikely to be discussed by guests, such as damage policy and trash policy. The topics specific for guest reviews are unlikely to be included in the other two types of documents, such as compliments like "perfect place" and complaints like "cancellation before arrival" as shown in Table K.3(b). In addition, we can learn topics about how hosts evaluate guests in their specific perspectives, such as guests following house rules and being fun talking to in Table K.3(c). The example topics further illustrate the validity of introducing shared and corpus-specific topics which can provide a more comprehensive understanding of the topic structure embedded in the different types of text inputs.

Table K.3. Examples of corpus-specific topics.

(a) Topics specific for item descriptions.

"damage policy"		**"trash policy"**	
property	0.0339	leave	0.0284
guest	0.0320	trash	0.0277
damage	0.0154	garbage	0.0202
owner	0.0150	recycle	0.0171
rental	0.0150	bin	0.0157

(b) Topics specific for guest reviews

"perfect place"		**"cancellation before arrival"**	
place	0.0537	arrival	0.0959
stay	0.0434	reservation	0.0887
perfect	0.0317	day	0.0849
beautiful	0.0217	cancel	0.0755
super	0.0216	posting	0.0675

(c) Topics specific for host reviews

"follow house rules"		**"fun talking"**	
house	0.3473	talk	0.1210
rule	0.2274	fun	0.1114
respect	0.0692	time	0.0451
follow	0.0672	great	0.0443
clean	0.0515	couple	0.0348

Appendix L. Details of Logistic Regression Results

In Table L.1, we display the entire list of topic-related features used in the Logistic Regression together with their coefficients, significance levels, and the coherence score and the percentage of documents covered by the corresponding topic. The vast majority of all the topic-related features has a significant coefficient, which means a significant effect on transaction success and failure. Moreover, we bold every topic-related feature with a significant coefficient and a decent coherence score (greater than 0.375), as illustrated in Table L.1.

Table L.1. Summary of Logistic Regression results.

#	Feature	Est.	Std. Error	Sign.	% of documents	Coherence
1	**"bar"**	**32.84**	**3.25**	***	**20.2%**	**0.875**
2	**"trash policy"**	**28.46**	**2.54**	***	**21.3%**	**0.75**
3	**"large area"**	**44.74**	**3.25**	***	**23.1%**	**0.875**
4	**"express ride"**	**21.21**	**3.02**	***	**24.9%**	**0.5**
5	**"share room"**	**30.93**	**3.51**	***	**34.3%**	**0.75**
6	**"mountain resort"**	**7.58**	**1.58**	***	**12.0%**	**0.875**
7	**"downtown"**	**28.70**	**3.01**	***	**25.6%**	**0.375**
8	**"street direction"**	**23.34**	**1.79**	***	**19.4%**	**0.625**
9	**"city apartment"**	**63.04**	**2.76**	***	**17.8%**	**0.5**
10	**"hot tub"**	**15.75**	**2.08**	***	**11.6%**	**0.5**
11	**"private room"**	**22.15**	**1.82**	***	**61.1%**	**0.75**
12	**"walking minutes to subway"**	**23.81**	**3.51**	***	**42.1%**	**0.625**
13	"artist neighborhood"	32.69	2.44	***	15.9%	0.25
14	"check-out policy"	30.43	3.42	***	19.9%	0.125
15	**"bike"**	**46.25**	**2.75**	***	**26.1%**	**0.5**
16	**"room & bed"**	**54.63**	**3.15**	***	**26.8%**	**0.375**
17	**"neighborhood & backyard"**	**52.16**	**3.22**	***	**33.5%**	**0.5**
18	**"airport transportation"**	**27.52**	**3.48**	***	**39.1%**	**0.75**
19	**"garden"**	**19.10**	**1.57**	***	**11.1%**	**0.875**
20	**"museum"**	**19.56**	**2.28**	***	**17.3%**	**0.75**
21	**"house"**	**35.65**	**3.03**	***	**22.4%**	**0.875**
22	**"towel & iron"**	**71.89**	**3.52**	***	**40.3%**	**0.5**
23	**"property damage"**	**17.98**	**3.42**	***	**19.7%**	**0.625**
24	**"walking minutes to station"**	**35.31**	**2.61**	***	**18.9%**	**0.5**
25	"location of the studio"	53.19	3.44	***	34.9%	0.125
26	**"check-in policy"**	**80.35**	**3.51**	***	**39.4%**	**0.375**
27	**"bedroom suite"**	**47.68**	**3.51**	***	**30.8%**	**0.75**
28	"local food"	23.86	2.26	***	15.2%	0.125

29	**“city centre & free parking”**	**22.08**	**2.10**	***	**17.9%**	**0.375**
30	“driving minutes”	36.32	2.94	***	19.1%	0.25
31	**“miles to park and downtown”**	**21.41**	**2.52**	***	**18.1%**	**0.75**
32	**“close to park”**	**15.09**	**3.04**	***	**24.9%**	**0.375**
33	**“train”**	**17.92**	**3.48**	***	**52.2%**	**0.5**
34	**“bed & breakfast”**	**21.46**	**1.90**	***	**13.8%**	**0.375**
35	**“door key”**	**54.64**	**2.98**	***	**32.7%**	**0.625**
36	**“loft”**	**27.82**	**3.19**	***	**43.0%**	**0.625**
37	**“park & museum”**	**26.68**	**3.47**	***	**37.7%**	**0.5**
38	**“health-related”**	**32.51**	**2.56**	***	**16.3%**	**0.375**
39	**“clean & nice”**	**35.03**	**2.56**	***	**38.1%**	**0.75**
40	**“island & ferry”**	**12.60**	**2.09**	***	**17.1%**	**0.625**
41	**“unique & great place”**	**53.65**	**2.43**	***	**31.7%**	**0.875**
42	**“block & neighborhood”**	**63.37**	**3.37**	***	**45.7%**	**0.75**
43	“time square”	41.28	3.42	***	39.3%	0
44	“condo units”	32.71	2.46	***	22.6%	0.125
45	**“seaside”**	**14.70**	**3.30**	***	**13.7%**	**0.625**
46	“cat friendly”	11.49	1.57	***	17.5%	0.25
47	**“room situation”**	**34.26**	**3.50**	***	**53.3%**	**1**
48	“great location”	1.51	2.02	0.448	27.8%	0.625
49	**“doors & windows”**	**81.67**	**3.11**	***	**35.2%**	**0.5**
50	**“stairs & floors”**	**25.53**	**1.52**	***	**23.7%**	**0.5**
51	**“east/west village”**	**20.92**	**3.47**	***	**29.4%**	**0.625**
52	“view”	46.14	3.31	***	31.8%	0.25
53	**“town”**	**24.70**	**1.78**	***	**20.1%**	**0.5**
54	**“well located”**	**47.63**	**3.50**	***	**55.5%**	**0.75**
55	**“subway”**	**30.86**	**2.82**	***	**38.7%**	**0.875**
56	**“cottage & cabin”**	**15.99**	**1.84**	***	**11.5%**	**0.875**
57	**“brownstone”**	**30.78**	**3.45**	***	**36.7%**	**0.625**
58	“breakfast”	37.55	2.82	***	26.9%	0.25
59	**“great place”**	**28.55**	**1.99**	***	**30.3%**	**0.375**
60	“garden”	21.41	1.26	***	18.6%	0.25
61	**“beautiful & wonderful stay”**	**71.74**	**3.51**	***	**33.7%**	**0.375**
62	“comfortable”	32.60	2.87	***	29.1%	0.25
63	“cancellation before arrival”	-6.12	3.51	.	26.8%	0.25
64	**“key location”**	**26.29**	**3.32**	***	**33.0%**	**0.5**
65	**“clean place”**	**21.81**	**2.61**	***	**30.0%**	**0.875**
66	**“breakfast”**	**56.41**	**3.19**	***	**23.3%**	**0.875**
67	**“great stay”**	**62.01**	**3.04**	***	**54.7%**	**0.5**

68	**"nice & good"**	**66.54**	**3.44**	***	**48.9%**	**0.5**
69	**"noise"**	**42.22**	**3.36**	***	**33.4%**	**0.5**
70	**"restaurant"**	**40.63**	**2.98**	***	**26.2%**	**0.875**
71	**"station"**	**18.41**	**3.40**	***	**37.0%**	**0.625**
72	**"lovely stay"**	**62.84**	**3.51**	***	**36.6%**	**0.5**
73	"clean subway"	40.28	3.37	***	47.1%	0.25
74	**"nice place"**	**55.06**	**3.52**	***	**49.8%**	**0.875**
75	**"great location"**	**81.06**	**3.50**	***	**54.7%**	**0.625**
76	**"great stay in Prospect Park area"**	**19.72**	**3.38**	***	**15.0%**	**0.375**
77	**"nice room"**	**30.14**	**3.45**	***	**34.8%**	**0.75**
78	**"perfect place"**	**51.26**	**3.51**	***	**29.3%**	**0.375**
79	**"clean room"**	**40.68**	**3.34**	***	**44.3%**	**0.625**
80	**"bar & restaurant"**	**37.85**	**3.51**	***	**37.2%**	**0.625**
81	"great location"	49.41	3.52	***	39.5%	0.25
82	**"central park location"**	**21.52**	**3.23**	***	**27.7%**	**0.875**
83	**"rooftop view"**	**35.31**	**3.45**	***	**21.3%**	**0.75**
84	**"airport"**	**40.01**	**3.32**	***	**28.9%**	**0.375**
85	**"great place"**	**29.71**	**2.51**	***	**60.7%**	**0.375**
86	**"loft"**	**35.78**	**2.90**	***	**12.5%**	**0.75**
87	**"studio location"**	**41.34**	**2.95**	***	**23.4%**	**0.625**
88	"dog-related"	20.06	2.38	***	13.8%	0
89	**"great family"**	**65.74**	**3.48**	***	**31.7%**	**0.5**
90	**"clean & nice"**	**59.65**	**3.49**	***	**50.7%**	**0.75**
91	**"island & ferry"**	**3.96**	**1.48**	*	**9.7%**	**0.625**
92	**"unique & great place"**	**32.48**	**2.83**	***	**21.5%**	**0.875**
93	**"block & neighborhood"**	**11.55**	**2.00**	***	**30.7%**	**0.75**
94	"time square"	11.94	1.83	***	23.7%	0
95	"condo units"	9.93	1.23	***	20.5%	0.125
96	**"seaside"**	**3.34**	**1.56**	*	**11.0%**	**0.625**
97	"cat friendly"	16.14	1.68	***	11.4%	0.25
98	**"room situation"**	**16.77**	**2.12**	***	**34.4%**	**1**
99	**"great location"**	**5.85**	**1.27**	***	**25.0%**	**0.625**
100	**"doors & windows"**	**24.08**	**2.73**	***	**32.7%**	**0.5**
101	**"stairs & floors"**	**28.74**	**3.49**	***	**33.9%**	**0.5**
102	**"east/west village"**	**12.41**	**1.67**	***	**15.1%**	**0.625**
103	"view"	11.92	1.51	***	15.0%	0.25
104	"town"	0.88	2.78	0.753	23.3%	0.5
105	**"well located"**	**12.48**	**3.05**	***	**54.6%**	**0.75**
106	**"subway"**	**15.34**	**3.21**	***	**41.5%**	**0.875**

107	"cottage & cabin"	1.34	1.18	0.257	11.5%	0.875
108	**"brownstone"**	**4.38**	**0.82**	***	**14.2%**	**0.625**
109	"breakfast"	7.38	0.97	***	18.9%	0.25
110	**"great place"**	**27.27**	**2.71**	***	**32.5%**	**0.375**
111	"garden"	45.88	2.76	***	20.3%	0.25
112	"highly recommend"	-3.57	2.55	0.159	39.5%	0.125
113	**"check-in & out"**	**21.11**	**3.18**	***	**46.6%**	**0.375**
114	**"smooth communication"**	**5.68**	**1.64**	**	**28.2%**	**0.75**
115	**"helpful & comfortable"**	**70.04**	**3.39**	***	**46.8%**	**0.625**
116	**"great experience"**	**46.37**	**3.41**	***	**36.3%**	**0.625**
117	**"stay time"**	**29.69**	**2.92**	***	**26.2%**	**0.875**
118	"neighborhood & family"	37.22	2.96	***	36.6%	0
119	**"great group"**	**32.39**	**3.47**	***	**33.6%**	**0.625**
120	"super clean"	-0.92	3.19	0.774	38.7%	0.25
121	**"happy experience"**	**59.82**	**3.46**	***	**37.3%**	**0.875**
122	**"family & friend"**	**63.25**	**3.25**	***	**32.2%**	**0.875**
123	**"leave it immaculate"**	**31.84**	**3.32**	***	**32.7%**	**0.375**
124	**"independent & quiet"**	**25.19**	**2.85**	***	**26.1%**	**0.625**
125	**"communicate well about plan"**	**40.30**	**3.19**	***	**35.7%**	**0.375**
126	**"polite & friendly"**	**29.39**	**2.58**	***	**32.6%**	**0.625**
127	"good to deal with"	24.93	2.82	***	28.7%	0.25
128	**"check-out"**	**33.31**	**3.52**	***	**32.5%**	**0.375**
129	**"in good order"**	**28.60**	**2.77**	***	**27.9%**	**0.625**
130	**"considerate & responsive"**	**31.85**	**2.24**	***	**28.4%**	**0.5**
131	**"awesome & amazing guest"**	**38.99**	**3.09**	***	**33.0%**	**0.5**
132	**"pleasure to host"**	**63.24**	**3.48**	***	**56.7%**	**0.625**
133	**"definitely recommend"**	**38.49**	**3.11**	***	**35.6%**	**0.625**
134	**"easy to communicate"**	**40.53**	**3.48**	***	**52.9%**	**0.375**
135	**"follow house rules"**	**31.60**	**3.51**	***	**32.5%**	**0.375**
136	**"duration of stay"**	**64.11**	**3.44**	***	**41.4%**	**1**
137	**"leave it spotless"**	**14.14**	**3.35**	***	**48.4%**	**0.625**
138	**"lovely couple"**	**50.60**	**2.66**	***	**30.7%**	**0.5**
139	**"enjoy guest visit"**	**62.34**	**3.45**	***	**34.4%**	**0.625**
140	**"friendly"**	**81.49**	**3.53**	***	**48.6%**	**0.625**
141	**"excellent guest"**	**17.02**	**2.49**	***	**27.4%**	**0.5**
142	**"good communicator"**	**36.51**	**2.54**	***	**38.9%**	**0.875**
143	"glad to host"	11.19	2.70	***	16.6%	0.125
144	**"delightful"**	**65.82**	**3.47**	***	**44.6%**	**0.875**
145	**"fun talking"**	**49.45**	**3.44**	***	**33.4%**	**0.75**

146	**“amazing people”**	**42.42**	**3.07**	***	**28.6%**	**0.5**
147	“clean & quiet”	16.47	1.62	***	22.3%	0.25
148	**“have guest anytime”**	**59.92**	**3.50**	***	**45.2%**	**0.375**
149	**“enjoyable conversation”**	**29.42**	**2.87**	***	**22.3%**	**0.5**
150	**“communicative”**	**50.32**	**3.26**	***	**35.8%**	**0.875**
151	“check-out”	2.00	2.40	0.406	13.0%	0.875
152	**“treatment”**	**12.67**	**1.69**	***	**17.0%**	**0.5**
153	**“great guest”**	**64.19**	**3.13**	***	**62.8%**	**0.375**
154	**“booking”**	**13.60**	**2.53**	***	**17.9%**	**0.375**
155	“short stay”	20.54	2.47	***	17.2%	0
156	**“kind”**	**36.44**	**2.48**	***	**35.9%**	**0.5**
157	**“considerate & friendly”**	**28.34**	**3.27**	***	**26.5%**	**0.375**
158	**“leave it in good shape”**	**38.46**	**3.52**	***	**45.8%**	**0.375**
159	“neat”	8.10	1.25	***	18.8%	0.125
160	“(no) hesitation”	1.83	1.85	0.326	16.3%	0
161	**“tidy”**	**49.30**	**3.40**	***	**38.8%**	**0.875**
162	**“happy communication”**	**13.93**	**2.24**	***	**16.1%**	**0.875**
163	**“visit city”**	**32.38**	**3.34**	***	**22.6%**	**1**
164	“recommend to future hosts”	44.14	3.25	***	29.4%	0.25
165	**“responsible”**	**18.22**	**2.18**	***	**22.6%**	**0.75**
166	“leave it in perfect condition”	41.64	3.15	***	34.0%	0.125
167	**“nice & clean”**	**44.09**	**3.06**	***	**57.8%**	**0.875**
168	“easy & flexible”	16.37	3.21	***	26.9%	0.125
169	**“respectful”**	**48.75**	**3.45**	***	**58.4%**	**0.75**
170	“absolutely recommend”	26.14	1.99	***	21.6%	0
171	**“quiet”**	**51.46**	**3.47**	***	**32.7%**	**1**
172	“highly recommend”	50.12	3.52	***	39.5%	0.125
173	**“check-in & out”**	**15.22**	**3.45**	***	**32.7%**	**0.375**
174	**“smooth communication”**	**49.89**	**3.51**	***	**39.3%**	**0.75**
175	**“helpful & comfortable”**	**7.93**	**1.17**	***	**17.3%**	**0.625**
176	**“great experience”**	**6.35**	**1.24**	***	**16.0%**	**0.625**
177	**“stay time”**	**12.68**	**1.91**	***	**17.5%**	**0.875**
178	“neighborhood & family”	3.67	0.78	***	12.6%	0
179	**“great group”**	**14.16**	**3.24**	***	**21.8%**	**0.625**
180	“super clean”	6.00	1.89	**	15.8%	0.25

Appendix M. Sensitivity Analysis

In order to further examine the robustness of our method, we conduct sensitivity analysis on the number of shared topics, the review history window and the hyper-parameter η used to generate word distributions, and the results demonstrate the robustness of our proposed model. Note that all the results presented here are based on BRTM-Sample.

The number of topics shared by listing descriptions and guest reviews is currently set as 22, and the number of topics shared by guest reviews and host reviews is set as 9. We vary these two parameters respectively within a small range. As illustrated in Table M.1, the hit rate of our proposed model is stable when the number of shared topics varies.

Table M.1. The hit rate as the number of shared topics varies.

(a) Varying the number of topics shared by *D* and *A*

Top-N	$T_{D\&A}$				
	18	**20**	**22**	**24**	**26**
1	0.209	0.213	**0.204**	0.208	0.203
2	0.371	0.373	**0.363**	0.366	0.361
3	0.504	0.502	**0.493**	0.496	0.493
4	0.613	0.611	**0.606**	0.609	0.604
5	0.706	0.704	**0.698**	0.702	0.698
6	0.782	0.779	**0.778**	0.781	0.774
7	0.841	0.841	**0.839**	0.840	0.834
8	0.888	0.887	**0.885**	0.885	0.881
9	0.921	0.921	**0.920**	0.920	0.917
10	0.945	0.945	**0.945**	0.944	0.943

(b) Varying the number of topics shared by *A* and *B*

Top-N	$T_{A\&B}$				
	7	**8**	**9**	**10**	**11**
1	0.203	0.207	**0.204**	0.205	0.207
2	0.359	0.361	**0.363**	0.362	0.364
3	0.489	0.493	**0.493**	0.492	0.498
4	0.602	0.605	**0.606**	0.604	0.611
5	0.696	0.699	**0.698**	0.699	0.702
6	0.773	0.776	**0.778**	0.775	0.781
7	0.835	0.837	**0.839**	0.836	0.842
8	0.881	0.883	**0.885**	0.886	0.888
9	0.918	0.920	**0.920**	0.920	0.922
10	0.942	0.945	**0.945**	0.945	0.946

Moreover, we conducted the history-window-size sensitivity analysis for both guest reviews and host reviews. As shown in Table M.2, using all available reviews still generates the best performance. Because the number of reviews a guest has written is relatively low based on Figure 6 in the main text, the performance using only the 10 most recent guest reviews is almost the same as that using all reviews. As for the host, when the number of recent host reviews used in the experiment reaches 15, the hit rate becomes almost the same as using all reviews.

Table M.2. The hit rate as the number of recent reviews used varies.

Top-N	BRTM-Sample	# of recent guest reviews				# of recent host reviews				
		1	3	5	10	1	3	5	10	15
1	**0.204**	0.168	0.200	0.202	0.208	0.165	0.193	0.201	0.199	0.202
2	**0.363**	0.307	0.355	0.354	0.364	0.286	0.343	0.357	0.355	0.357
3	**0.493**	0.424	0.484	0.482	0.492	0.390	0.472	0.488	0.482	0.489
4	**0.606**	0.533	0.593	0.594	0.603	0.485	0.585	0.599	0.594	0.599
5	**0.698**	0.627	0.689	0.688	0.696	0.576	0.682	0.695	0.690	0.695
6	**0.778**	0.711	0.765	0.765	0.770	0.664	0.762	0.774	0.767	0.774
7	**0.839**	0.781	0.829	0.827	0.831	0.741	0.826	0.836	0.830	0.837
8	**0.885**	0.839	0.878	0.877	0.882	0.809	0.876	0.883	0.880	0.883
9	**0.920**	0.887	0.913	0.915	0.918	0.866	0.913	0.919	0.916	0.920
10	**0.945**	0.921	0.939	0.939	0.944	0.910	0.941	0.944	0.943	0.945

In addition, we vary the value of the hyper-parameter η used to generate word distributions, and the hit rate remains stable as shown in Table M.3. The value we adopted in our experiments is 0.01.

Table M.3. The hit rate as the hyper-parameter η varies.

Top-N	Hyper-parameter η				
	0.0001	0.001	0.01	0.1	1
1	0.206	0.211	**0.204**	0.202	0.190
2	0.366	0.372	**0.363**	0.359	0.344
3	0.498	0.505	**0.493**	0.492	0.471
4	0.609	0.615	**0.606**	0.605	0.578
5	0.702	0.707	**0.698**	0.700	0.673
6	0.780	0.782	**0.778**	0.775	0.751
7	0.841	0.844	**0.839**	0.837	0.817
8	0.888	0.889	**0.885**	0.884	0.869
9	0.922	0.923	**0.920**	0.919	0.908
10	0.946	0.948	**0.945**	0.945	0.937

Appendix N. Results without Instant Bookings

For further robust check, we conduct experiments on the data containing no instant bookings. As displayed in Table N.1, our proposed model still performs the best among all the alternative methods.

Table N.1. Comparison of hit rate with representative baselines excluding instant bookings.

Top-N	RAND	RANK-RATING	CF-G	LDA-G	DL-CNN	RTM-G	RTM-GH	BRTM-SEP	BRTM-Sample
1	0.050 (+300.0%)	0.067 (+197.5%)	0.104 (+92.3%)	0.124 (+61.3%)	0.168 (+18.8%)	0.110 (+81.5%)	0.174 (+14.7%)	0.179 (+11.4%)	**0.200**
2	0.100 (+259.2%)	0.142 (+152.0%)	0.185 (+93.8%)	0.239 (+50.4%)	0.302 (+18.7%)	0.215 (+67.2%)	0.315 (+14.0%)	0.324 (+10.9%)	**0.359**
3	0.149 (+233.7%)	0.228 (+118.0%)	0.259 (+91.9%)	0.340 (+46.1%)	0.419 (+18.7%)	0.312 (+59.3%)	0.434 (+14.5%)	0.443 (+12.1%)	**0.497**
4	0.199 (+208.0%)	0.315 (+94.7%)	0.328 (+87.1%)	0.436 (+40.6%)	0.513 (+19.6%)	0.405 (+51.4%)	0.543 (+13.0%)	0.548 (+12.0%)	**0.613**
5	0.247 (+187.5%)	0.400 (+77.6%)	0.393 (+80.7%)	0.515 (+37.8%)	0.595 (+19.4%)	0.490 (+45.0%)	0.631 (+12.6%)	0.636 (+11.7%)	**0.710**
6	0.297 (+165.8%)	0.478 (+65.1%)	0.457 (+72.6%)	0.593 (+33.1%)	0.660 (+19.6%)	0.571 (+38.2%)	0.705 (+11.9%)	0.707 (+11.6%)	**0.789**
7	0.346 (+145.1%)	0.552 (+53.8%)	0.519 (+63.5%)	0.662 (+28.1%)	0.717 (+18.3%)	0.641 (+32.3%)	0.764 (+11.1%)	0.768 (+10.4%)	**0.848**
8	0.398 (+124.2%)	0.617 (+44.5%)	0.576 (+54.7%)	0.720 (+23.9%)	0.764 (+16.7%)	0.707 (+26.1%)	0.813 (+9.68%)	0.817 (+9.16%)	**0.892**
9	0.449 (+105.5%)	0.673 (+37.2%)	0.632 (+45.9%)	0.772 (+19.5%)	0.806 (+14.5%)	0.765 (+20.5%)	0.853 (+8.17%)	0.856 (+7.84%)	**0.923**
10	0.500 (+89.0%)	0.720 (+31.2%)	0.687 (+37.4%)	0.815 (+15.9%)	0.840 (+12.4%)	0.816 (+15.7%)	0.886 (+6.59%)	0.887 (+6.46%)	**0.944**

Appendix O. Experiments on More Datasets

For further validation of the effectiveness of our proposed model, we conduct experiments on another Airbnb dataset about London and a dataset from a boat sharing platform called Boatsetter for robustness check. The increase of the predictive performance still holds in experiments on both datasets.

O.1 Experiments on the Airbnb-London Dataset

Following the same process of building the Airbnb-NYC dataset, we collect all the information relevant to the listings located in London up to June 30th, 2019, including listing descriptions, bilateral reviews, and user profiles. Descriptive statistics of our Airbnb-London dataset are provided in Table O.1.

Table O.1. Summary of the Airbnb-London dataset.

#Guests	#Hosts	#Listings	#Transactions in one year	#Reviews written by all guests	#Reviews written by all hosts
601,947	41,931	73,392	716,208	2,864,511	2,922,081

We split the transactions with reviews in one year (from July 1st, 2018 to June 30th, 2019) into three subsets: the training set (the first eight months), the validation set (the next one month), and the test set (the last three months). Likewise, we determine proper parameters for our proposed BRTM model based on performances on the validation set. The number of corpus-specific topics for listing descriptions, guest reviews and host reviews are T_{D^*} = 40, T_{A^*} = 29, and T_{B^*} = 49 respectively, and the number of shared topics between guest reviews and item descriptions is set as $T_{D\&A}$ = 20 and that between guest reviews and host reviews is $T_{A\&B}$ = 11. Other experimental settings keep unchanged. As displayed in Table O.2, our proposed BRTM model (using BRTM-Sample as a representative) still performs the best among all the alternatives. Compared to RTM-GH and BRTM-SEP, our BRTM model can achieve a 10.0% increase in the hit rate of top-N recommendation.

Table O.2. Comparison of Hit Rate on the Airbnb-London dataset.

Top-N	RAND	RANK-RATING	CF-G	LDA-G	DL-CNN	RTM-G	RTM-GH	BRTM-SEP	BRTM
1	0.050 (+317.1%)	0.097 (+116.0%)	0.161 (+30.1%)	0.163 (+28.3%)	0.163 (+29.0%)	0.162 (+29.5%)	0.189 (+10.8%)	0.182 (+15.1%)	**0.210**
2	0.097 (+250.3%)	0.182 (+87.5%)	0.265 (+28.8%)	0.292 (+16.8%)	0.294 (+16.0%)	0.286 (+19.6%)	0.309 (+10.4%)	0.297 (+15.0%)	**0.341**
3	0.148 (+204.5%)	0.249 (+80.3%)	0.349 (+28.7%)	0.402 (+11.8%)	0.401 (+12.2%)	0.389 (+15.5%)	0.403 (+11.6%)	0.391 (+14.9%)	**0.450**
4	0.196 (+172.7%)	0.308 (+74.0%)	0.424 (+26.5%)	0.492 (+8.83%)	0.495 (+8.33%)	0.483 (+11.0%)	0.484 (+10.7%)	0.476 (+12.5%)	**0.536**
5	0.244 (+151.7%)	0.362 (+69.4%)	0.494 (+24.2%)	0.575 (+6.72%)	0.578 (+6.25%)	0.565 (+8.62%)	0.557 (+10.2%)	0.548 (+12.1%)	**0.614**

O.2 Experiments on the Boatsetter Dataset

We conduct additional experiments on another real-world dataset from Boatsetter, which offers a boat sharing marketplace, a.k.a, "Airbnb for boats". On this platform, boat descriptions and bilateral reviews are important information sources for the two-stage decision-making process. We collect the data relevant to all the boats operated in the U.S. up to the end of 2019, including boat descriptions, reviews written by all customers, and reviews written by all boat owners. Descriptive statistics of the Boatsetter dataset are provided in Table O.3.

Table O.3. Summary of the Boatsetter dataset.

#Customers	#Boat owners	#Boats	#Transactions in one year	#Reviews written by all customers	#Reviews written by all boat owners
9,277	3,330	4,020	5,223	13,336	9,519

Following the similar setting in the experiments on Airbnb dataset, all the transactions in 2019 on Boatsetter are split into three parts: transactions in the first eight months for training, transactions in the next one month for validation and parameter selection, and transactions in the last three months for performance evaluation. Our proposed BRTM model can be directly implemented without any modification. Following similar process for parameter selection, the number of corpus-specific topics for boat descriptions, reviews by customers, and reviews by boat owners are T_{D^*} = 12, T_{A^*} = 9, and T_{B^*} = 12 respectively, and the number of shared topics between customer reviews and boat descriptions is set as $T_{D\&A}$ = 3 and that between customer reviews and owner reviews is $T_{A\&B}$ = 3. Other experimental settings keep unchanged. In Table O.4, we use BRTM-Sample as the representative method and present the comparison of the hit rate with other alternatives. As we can see, our proposed BRTM model still outperforms all the other baselines.

Table O.4. Comparison of Hit Rate on the Boatsetter dataset.

Top-N	RAND	RANK-RATING	CF-G	LDA-G	RTM-G	RTM-GH	BRTM-SEP	BRTM
1	0.049 (+496.6%)	0.099 (+193.2%)	0.117 (+147.1%)	0.157 (+84.0%)	0.168 (+73.0%)	0.241 (+20.1%)	0.236 (+22.7%)	**0.290**
2	0.094 (+371.4%)	0.245 (+80.8%)	0.157 (+180.9%)	0.288 (+53.5%)	0.358 (+23.4%)	0.395 (+11.9%)	0.389 (+13.8%)	**0.442**
3	0.154 (+270.7%)	0.347 (+64.7%)	0.221 (+158.3%)	0.424 (+34.8%)	0.524 (+8.95%)	0.533 (+7.23%)	0.534 (+6.90%)	**0.571**
4	0.211 (+215.9%)	0.462 (+44.2%)	0.285 (+134.1%)	0.558 (+19.5%)	0.631 (+5.57%)	0.658 (+1.27%)	0.648 (+2.84%)	**0.667**
5	0.260 (+186.5%)	0.554 (+34.1%)	0.343 (+116.6%)	0.647 (+15.0%)	0.734 (+1.37%)	0.740 (+0.45%)	0.747 (-0.45%)	**0.744**

Appendix P. List of Platforms with Bilateral Reviews

Our BRTM framework can be applied to platforms where bilateral reviews play an important role in user decision making. In recent years, more and more platforms offering bilateral reviews have emerged. Below we present a list of platforms with bilateral reviews in different fields. Among all these platforms, Airbnb, BoatSetter, Peerspace, and Splacer provide public access to bilateral reviews for all users on the platform. Other platforms only allow the public to see reviews written by the buyers or even hold all bilateral reviews private unless the user is involved in a potential transaction.

Accommodation

- Airbnb: A leading marketplace for accommodation sharing. It now also offers experience sharing.
- HomeAway: A popular vacation rental marketplace.
- Love Home Swap: A home-swapping website enabling users to exchange and rent homes.

Space Sharing

- Peerspace: A peer-to-peer marketplace for booking space for events, meetings, and productions.
- Splacer: An online platform for renting space for birthday parties, weddings, and corporate events.

Vehicle Renting

- BoatSetter: A boat sharing marketplace.
- Getaround: An online platform enabling private car owners to rent out their cars.
- Turo: An online platform providing car sharing services.

Delivery and Logistics

- uShip: An online marketplace offering fast shipping services of large items, including auto transport, boat shipping, moving services, and so on.

Dining

- EatWith: An online platform bringing locals and travelers together to share food experiences.
- MealSharing: Another meal sharing platform.

Freelance

- Fiverr: An online marketplace for freelance services, such as graphic design, translation, programming and so on.
- Handy: An online marketplace for house/home cleaning and handyman services.

- TaskRabbit: An online marketplace that connects freelance labor with local demand, such as personal assistance, office service, and many other handyman services.
- Bidvine: An online platform that helps find local service professionals.

References

Abrahams, A. S., Jiao, J., Wang, G. A., & Fan, W. (2012). Vehicle defect discovery from social media. Decision Support Systems, 54(1):87-97.

Blei, D. M., & Jordan, M. I. (2006). Variational inference for Dirichlet process mixtures. Bayesian Analysis, 1(1):121-143.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. J. Machine Learn. Res., 3:993-1022.

Braun, M., & McAuliffe, J. (2010). Variational inference for large-scale models of discrete choice. J. American Statistical Association, 105(489):324-335.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. J. Machine Learn. Res., 2493-2537.

Chang, J., & Blei, D. M. (2010). Hierarchical relational models for document networks. The Annals of Applied Statist., 124-150.

Chang, J., Gerrish, S., Wang, C., Boyd-Graber, J. L., & Blei, D. M. (2009). Reading tea leaves: How humans interpret topic models. Adv. Neural Inform. Processing Systems (NIPS), 288-296.

He, X., Zhang, H., Kan, M. Y., & Chua, T. S. (2016). Fast matrix factorization for online recommendation with implicit feedback. Proc. 39th Int'l ACM SIGIR Conf. Research and Development in Information Retrieval, 549-558.

Hu, Y., Koren, Y., & Volinsky, C. (2008). Collaborative filtering for implicit feedback datasets. Proc. 8th IEEE Int'l Conf. Data Mining, 263-272.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., & Saul, L. K. (1999). An introduction to variational methods for graphical models. Machine Learning, 37(2):183-233.

Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. Proc. 2014 Conf. Empirical Methods in Natural Language Processing (EMNLP), 1746-1751.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Adv. Neural Inform. Processing Systems (NIPS), 1097-1105.

Loria, S. (2018). TextBlob Documentation. Release 0.15.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013a). Efficient estimation of word representations in vector space. ArXiv preprint, arXiv:1301.3781.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. Adv. Neural Inform. Processing Systems (NIPS), 3111-3119.

Miller, G. A. (1998). WordNet: An electronic lexical database. MIT Press.

Minsky, M. L., & Papert, S. (1969). Perceptrons: An introduction to computational geometry.

Pan, R., & Scholz, M. (2009). Mind the gaps: Weighting the unknown in large-scale one-class collaborative filtering. Proc. 15th ACM SIGKDD Int'l Conf. Knowled. Discov. and Data Mining, 667-676.

Pan, R., Zhou, Y., Cao, B., Liu, N. N., Lukose, R., Scholz, M., & Yang, Q. (2008). One-class collaborative filtering. Proc. 8th IEEE Int'l Conf. Data Mining, 502-511.

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. Proc. 2014 Conf. Empirical Methods in Natural Language Processing (EMNLP), 1532-1543.

Rendle, S., Freudenthaler, C., Gantner, Z., & Schmidt-Thieme, L. (2009). BPR: Bayesian personalized ranking from implicit feedback. Proc. 25th Conf. Uncertainty in Artificial Intelligence, 452-461.

Salton, G., Wong, A., & Yang, C. S. (1975). A vector space model for automatic indexing. Communications of the ACM, 18(11):613-620.

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. Proc. 10th Int'l Conf. World Wide Web, 285-295.

Zhang, L., Wang, S., & Liu, B. (2018). Deep learning for sentiment analysis: A survey. Wiley Interdisciplinary Reviews: Data Mining and Knowled. Discovery, 8(4):e1253.

Zhu, C., Byrd, R. H., Lu, P., & Nocedal, J. (1997). Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. ACM Tran. Mathematical Software (TOMS), 23(4), 550-560.