Multimodal Review Generation for Recommender Systems

Quoc-Tuan Truong
Singapore Management University
Singapore
qttruong.2017@smu.edu.sg

Hady W. Lauw
Singapore Management University
Singapore
hadywlauw@smu.edu.sg

ABSTRACT

Key to recommender systems is learning user preferences, which are expressed through various modalities. In online reviews, for instance, this manifests in numerical rating, textual content, as well as visual images. In this work, we hypothesize that modelling these modalities jointly would result in a more holistic representation of a review towards more accurate recommendations. Therefore, we propose Multimodal Review Generation (MRG), a neural approach that simultaneously models a rating prediction component and a review text generation component. We hypothesize that the shared user and item representations would augment the rating prediction with richer information from review text, while sensitizing the generated review text to sentiment features based on user and item of interest. Moreover, when review photos are available, visual features could inform the review text generation further. Comprehensive experiments on real-life datasets from several major US cities show that the proposed model outperforms comparable multimodal baselines, while an ablation analysis establishes the relative contributions of the respective components of the joint model.

ACM Reference Format:
Quoc-Tuan Truong and Hady W. Lauw. 2019. Multimodal Review Generation for Recommender Systems. In *Proceedings of the 2019 World Wide Web Conference (WWW '19), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3308558.3313463

1 INTRODUCTION

One classic formulation for recommender system is rating prediction [13]. Increasingly ratings are no longer the sole modality for expressing one's preference. The most common scenario now is when rating is but a component of an online review, which also contains a textual description of one's experiences with a product or a place. Figure 1 illustrates a review from Yelp.com for *Antica Ristorante*, a popular Italian restaurant in the Financial District of NYC. The writer *Adam W.* assigned it the highest rating of 5 stars, and articulated his impression through sentences stating supporting factors such as amazing and flavorful clam linguini, attentive servers, and homemade grappa. Evident from the example is how vividly the textual description embodies multi-faceted user's preferences, with the rating being a culminating assessment.

We start off with the premise that the textual content of a review would be useful to rating prediction. Upon perusing the existing literature on content-based recommender systems (see Section 2),

Figure 1: Example of *Yelp* review for Antica Ristorante

we observe a couple of distinctions in our intended setting here. First, when textual content was used for recommendation, in most cases it was in association with the item (e.g., product description, paper abstract), rather than with the user-item tuple (i.e., a review). Even in those cases that relied on review texts, the approach was usually to aggregate all the reviews belonging to an item into a document [19, 20]. Second, textual content was commonly seen as observations or features, rather than as a generative output.

Problem We express the problem as *multimodal review generation*. Given a user and an item, we seek to generate both the rating and the review text. Prior works tend to address one given the other. For instance, several content-based recommendations [10, 33] generate ratings based on observed text. On the other hand, product review generation [5] seeks to generate the review text based on user, item, and the observed rating. To showcase the benefits of joint modeling, we will compare to these classes of baselines. For an additional benefit, the generated text could also potentially serve as explanations [7, 37] to the rating-based recommendation.

Moreover, multimodality for a review goes beyond numerical rating and textual content. Given the proliferation of multimedia smartphones, arming virtually everyone with a camera on hand, online reviews frequently also contain photos. When such photos are available, they often provide an illustrative backdrop to the various facets mentioned in the text, as attested to by Figure 1. Hence, we hypothesize that photos could be helpful to review generation, and consider review photos as another input to the problem.

Approach We propose a model called *Multimodal Review Generation* or *MRG*, which joins rating prediction with non-linearity and review text generation with LSTM. The joint modeling implies that the users and items' representations would be sensitive to the text generation process, and the generated text would be personalized and item-sensitive. Another novel aspect to *MRG* is incorporating visual features in the text generation. The selection of words to describe an item is challenging given the large variance among items.

WWW '19, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6674-8/19/05.
https://doi.org/10.1145/3308558.3313463

To supplement the sentiment features encoded in users and items' representations, we will use visual features extracted from photos to provide additional clues. Intuitively, words that could describe a photo in a review would be suitable to describe some facet of the item of interest. However, we make the deliberate design decision to use image features to help generate review text, rather than to also attempt generating the images themselves. This is because our primary intention is to model user's subjective preferences, which would be expressed much more evidently in the rating and the text.

Contributions We design the *MRG* model (see Section 3), which jointly models rating prediction and text generation at the review level by incorporating LSTM cells with a novel fusion gate as a kind of soft attention to weigh the relative contributions of sentiment features and visual features that provide context to the text generation. We also describe the learning and inference algorithms respectively. In Section 4 we conduct comprehensive experiments against baselines for both rating prediction and review generation that showcase the benefits of their joint modeling and incorporation of visual features, as well as an ablation analysis that establishes the contributions of the various components of our model.

2 RELATED WORK

In this section, we review the two groups of literature most related to our problem, namely: content-based recommendation and text generation (particularly relating to online reviews).

Content-Based Recommendation. The vast majority of recommendation works are based on modeling ratings [31], and most rely on matrix factorization [13]. These include the popular methods such as Probabilistic Matrix Factorization or PMF[22], Non-negative Matrix Factorization or NMF [15], and Singular Value Decomposition with neighborhood information or SVD++ [12], which we will use as baselines to validate the benefits of content modeling. Note that we focus on models that fit rating values, rather than pairwise rankings [8, 29] that are generally seen as a different formulation.

Later approaches seek to model textual content to supplement ratings. One direction is to apply topic modeling to item content, and relate an item's topic distribution to its latent factors for use in matrix factorization-like rating prediction. Such approaches [19, 20, 32] outperform rating-only methods. The more recent direction is to rely on neural approaches. For instance, [33] improves upon [32] by replacing the topic model with a Stacked Denoising AutoEncoder (SDAE) to learn item latent factors. ConvMF [10] learns a Convolutional Neural Network to extract latent features for an item from its text reviews. DeepCoNN [38] models user and item's review texts separately using siamese CNN and joins the two sets of features in the manner of Factorization Machine [28]. The latter three are state-of-the-art, having been compared favorably to the prior ones based on topic models, and will be used as baselines.

There are two key distinctions between ours and the approaches in this category. First, we model review-level textual content, i.e., the text is associated with the user-item tuple, rather than with the item or the user alone. Second, we seek to generate a piece of natural language text as an output, as opposed to feature vectors.

Review Text Generation. Learning to generate text is intensively investigated in natural language processing, where it can be applied to language modeling, machine translation, or speech recognition. The idea is to learn a model which can generate a sequence of text based on sequential dependencies and contexts. Long Short-Term Memory (LSTM) [9], a gated version of recurrent neural networks, is capable of capturing long-term dependencies and is demonstrably an effective approach [3, 6, 31].

The review text generation technique closest to ours is Att2Seq [5]. The work models user, item, and rating as given attributes, and learns to generate a review text through an attention mechanism. Note that we do not take the rating as a given; instead we seek to generate the rating concurrently. In Section 4, we will compare to Att2Seq as one of the review generation baselines, assessing their performance in both scenarios when the test rating is unknown (our setting) and when it is given (their original setting). Other review generation methods are not directly comparable. For instance, some expect extraneous inputs not applicable to our setting, such as generating tips based on a review [16], generating review based on pre-defined phrases [25], or considering the neighborhood reviews of a user [34]. In turn, [24] is based on pairwise ranking, which is usually seen as a different formulation from rating prediction. None makes use of visual features from images.

By incorporating visual features in the text generation, our problem is related to image captioning [35, 36]. The key distinction is our incorporation of user and item latent factors, which implies that in our case the "captioning" is effectively item-sensitive and personalized. In Section 4 we will compare to [35] as a image-based text generation representative not affected by either user or item.

Table 1: List of notations

Symbols	Description
$\mathcal{X}^r$	set of rating observations
$\mathcal{X}^s$	set of review text with images
$\mathcal{U}$	set of users
$\mathcal{I}$	set of items
$\mathcal{A}$	set of image annotations
$\mathbf{P}$	user embedding matrix
$\mathbf{Q}$	item embedding matrix
$\mathbf{E}$	word embedding matrix
$\mathbf{W}$	weight projection matrix
b	bias term
Θ	set of neural parameters

3 MULTIMODAL REVIEW GENERATION

We now describe our proposed *Multimodal Review Generation* or *MRG*, first the model architecture, then the learning and inference. Table 1 reproduces the notations for ease of reference.

Problem. The universal sets of users and items are denoted $\mathcal{U}$ and $\mathcal{I}$. We are given $\mathcal{X}^r$ and $\mathcal{X}^s$ for learning. $\mathcal{X}^r$ is the set of rating observations, whereby each observation is a triple (u, i, r) indicating that a user $u \in \mathcal{U}$ assigns to an item $i \in \mathcal{I}$ a rating score of $r \in \mathbb{R}$. In turn, $\mathcal{X}^s$ is the set of review content observations, whereby each observation is a quadruple (u, i, d, m) with d being a review text document and m is a corresponding image. We further assume that for each observation in $\mathcal{X}^r$, there is at least one corresponding observation in $\mathcal{X}^s$ for the same user and item, i.e., each rating is accompanied by at least a review text and an image.

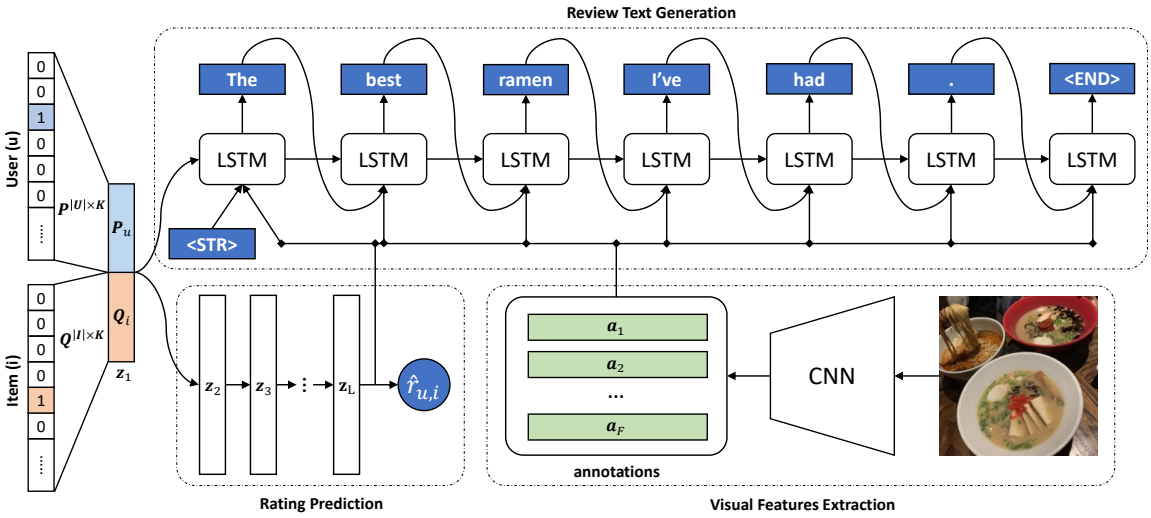


Figure 2: Overall Architecture of Multimodal Review Generation (MRG) model

The problem of multimodal review generation can thus be stated as follows. From $\mathcal{X}^r$ and $\mathcal{X}^s$, we seek to learn a model, which could output a predicted rating score $\hat{r}$ and a synthesized review document $\hat{d}$ given a user u, an item i, and an image m. The image is not a must, but could be useful. One use case is when a user is observed to have an image of an item (such as an Instagram posting), and we seek to assess the likely rating and review text. Alternatively, we could maintain a set of images reflecting different facets of an item, which could then be sampled for a given user. As we will see shortly, even when image is unavailable, the model is still capable of producing both a rating prediction and a synthesized review text.

3.1 Model Architecture

Fig. 2 shows the neural model architecture, with several components including rating prediction and review text generation, as well as visual features extraction. We describe each component in turn.

Rating Prediction with Non-Linearity. Traditionally, matrix factorization [13] models rating via a *bilinear* function, such as:

$$\hat{r}_{u,i} = \mathbf{p}_u^T \mathbf{q}_i + b_u + b_i + \mu \tag{1}$$

where $\mathbf{p}_u, \mathbf{q}_i$ are user and item latent factors, b_u, b_i are user and item biases, and μ is the global bias (average rating).

An emerging idea is to use *non-linearity* to capture the interactions between users and items. Non-linear transformations promise to learn better representations as shown for various tasks [6, 14, 21]. We propose to utilize Multilayer Perceptron or MLP as the rating prediction component. In a different formulation, MLP was combined with matrix factorization to learn pairwise ranking scores [8].

User u and item i are encoded as one-hot vectors. With the help of user embedding matrix $\mathbf{P}^{|\mathcal{U}|xK}$ and item embedding matrix $\mathbf{Q}^{|\mathcal{I}|xK}$, we project the one-hot vectors onto the user embedding vector $\mathbf{P}_u$ and item embedding vector $\mathbf{Q}_i$. These are concatenated and projected through a number L of neural layers followed by non-linear transformations, as shown below. The model outputs a rating $\hat{r}_{u,i}$ based on the representation learned from the last layer.

$$\mathbf{z}_1 = \begin{pmatrix} \mathbf{P}_u \\ \mathbf{Q}_i \end{pmatrix} \tag{2}$$

$$\mathbf{z}_2 = \phi(\mathbf{W}_2 \mathbf{z}_1 + b_2) \tag{3}$$

$$\ldots \tag{4}$$

$$\mathbf{z}_L = \phi(\mathbf{W}_L \mathbf{z}_{L-1} + b_L) \tag{5}$$

$$\hat{r}_{u,i} = \mathbf{W}_r \mathbf{z}_L + b_r \tag{6}$$

Above, $\mathbf{W}_l$ is learned projection matrix, b_l is bias of the neural layer l. In turn, ϕ is hyperbolic tangent ($tanh$) function, and $\mathbf{W}_r, b_r$ are projection matrix and bias of the output layer.

Review Text Generation with Sensitivity to User and Item. The text generation component (upper part of Fig. 2) is built on LSTM. First, we describe how we generate text as a language model with only textual contexts. Then, we introduce the notion of user and item with sentiment information from the rating prediction component. Later on, we extend it further with visual information.

Let $d = \{w_1, \ldots, w_T\}$ be a review text document, where w_t is the word at position t. We encode each word w_t as an one-hot vector $\mathbf{y}_t \in \mathbb{R}^V$, where V is the vocabulary size. Thus, document d becomes a sequence of one-hot vectors $\mathbf{y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_T\}$. We implement our LSTM (see Fig. 3) according to following equations:

$$\mathbf{x}_t = \mathbf{E}\mathbf{y}_{t-1} \tag{7}$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_{fx}\mathbf{x}_t + \mathbf{W}_{fh}\mathbf{h}_{t-1} + b_f) \tag{8}$$

$$\mathbf{i}_t = \sigma(\mathbf{W}_{ix}\mathbf{x}_t + \mathbf{W}_{ih}\mathbf{h}_{t-1} + b_i) \tag{9}$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_{ox}\mathbf{x}_t + \mathbf{W}_{oh}\mathbf{h}_{t-1} + b_o) \tag{10}$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \phi(\mathbf{W}_{cx}\mathbf{x}_t + \mathbf{W}_{ch}\mathbf{h}_{t-1} + b_c) \tag{11}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \phi(\mathbf{c}_t) \tag{12}$$

where $\mathbf{E} \in \mathbb{R}^{mxV}$ is the learned word embedding matrix initialized randomly or from pre-trained word embeddings [21, 27], $\mathbf{W}_*, b_*$ are learned projection matrices and biases initialized randomly, and σ, ϕ are *sigmoid* and *tanh* activation functions respectively.

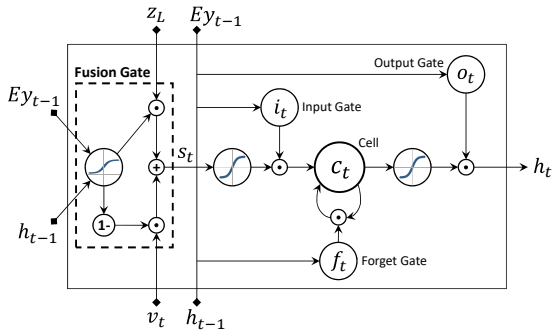


Figure 3: LSTM Cell with Fusion Gate

Because we seek to generate a document that would be relevant to the item at hand yet sensitive to user preferences, we initialize the LSTM with user embedding $\mathbf{P}_u$ and item embedding $\mathbf{Q}_i$.

$$\mathbf{c}_0 = \phi(\mathbf{W}_{c0}\mathbf{z}_1 + b_{c0}) \tag{13}$$

$$\mathbf{h}_0 = \phi(\mathbf{W}_{h0}\mathbf{z}_1 + b_{h0}) \tag{14}$$

Above, $\mathbf{z}_1$ is the concatenated vector of $\mathbf{P}_u$ and $\mathbf{Q}_i$ (Eq. 2). $\mathbf{W}_{c0}, \mathbf{W}_{h0} \in \mathbb{R}^{n\times 2K}$ and b_{c0}, b_{h0} are learned projection matrices and biases. n is the number of hidden dimensions of the LSTM state.

By sharing the user and item embeddings between the rating prediction and review text generation components, we seek shared representations that will be beneficial to both components [2, 23]. Moreover, in addition to sequential dependencies, we supply the LSTM with sentiment information. In other words, at each time step t, in addition to the generated word from $t-1$, we use $\mathbf{z}_L$ (Eq. 5) as another input source of the LSTM by simply concatenating it with previous generated word embedding:

$$\mathbf{x}_t = \begin{pmatrix} \mathbf{E}\mathbf{y}_{t-1} \\ \mathbf{z}_L \end{pmatrix} \tag{15}$$

The intuition to use $\mathbf{z}_L$ is because the final representation of the rating prediction component is hypothesized to be the most representative features of the sentiment of the review.

Review Text Generation with Visual Features. Based on the previously described components, we would already meet the dual objectives of rating prediction and review text generation. In some cases, additional information in the form of photo(s) may be available. For instance, a user may not have reviewed an item, but she may have an Instagram posting about the item. In such cases, we hypothesize that visual features would provide additional clues that could improve the review text generation further.

To extract feature vectors, also referred to as *annotations*, from images, we use convolutional neural network or CNN. For each image m, we get a set of annotation vectors $\mathcal{A} = \{\mathbf{a}_1, \ldots, \mathbf{a}_F\}, a_j \in \mathbb{R}^D$, where F is the number of annotations.

$$\mathcal{A} = \text{CNN}(m) \tag{16}$$

Our *MRG* architecture employs 19-layer Oxford VGGNet [30] as the visual feature extractor. We use the pre-trained model of VGGNet on ImageNet [4] without finetuning, and treat the $14 \times 14 \times 512$ feature map output from the fifth convolutional layer (conv5_3) as our image annotations. In consequence, each image will be transformed into 196 annotation vectors $\mathbf{a}_j$ with $D = 512$.

At each time step t, we want to supply our LSTM a context vector as a visual clue for generating the next word which is relevant to the image. Each annotation $\mathbf{a}_j$ is a compression of a small region of the image. Not all annotations are contributing visual information equally. Therefore, we apply soft attention mechanism [3, 35] to allow the model to weigh on the more relevant annotations. Thus, the visual context vector $\mathbf{v}_t$ is obtained by the following equations:

$$\mathbf{e}_{tj} = \phi(\mathbf{W}_{ea}\mathbf{a}_j + \mathbf{W}_{eh}\mathbf{h}_{t-1} + b_e) \tag{17}$$

$$\alpha_{tj} = \frac{\exp(\mathbf{W}_\alpha \mathbf{e}_{tj})}{\sum_{k=1}^{F}\exp(\mathbf{W}_\alpha \mathbf{e}_{tk})} \tag{18}$$

$$\mathbf{v}_t = \sum_{j=1}^{F} \alpha_{tj}\mathbf{a}_j \tag{19}$$

where $\mathbf{W}_{ea} \in \mathbb{R}^{D\times D}, \mathbf{W}_{eh} \in \mathbb{R}^{D\times n}, \mathbf{W}_\alpha \in \mathbb{R}^{1\times D}$, and b_e are learned projection matrices and bias randomly initialized.

Fusion Gate as Soft Attention between Sentiment and Visual Information. In addition to $\mathbf{z}_L$, the visual context vector $\mathbf{v}_t$ is treated as another source of input for our LSTM. Although we hypothesize that each generated word is either based on sentiment or visual information, we relax that constraint by allowing the model to make soft decisions. We introduce a fusion gate (Fig. 3), which can be seen as an attention mechanism for the LSTM. The fusion gate will learn to give the importance weights for sentiment features $\mathbf{z}_L$ and visual features $\mathbf{v}_t$ and combine them into a final context vector $\mathbf{s}_t$. More information from visual features means less information from sentiment features going through the gate.

$$\gamma_t = \sigma(\mathbf{W}_{\gamma y}\mathbf{E}\mathbf{y}_{t-1} + \mathbf{W}_{\gamma h}\mathbf{h}_{t-1} + b_\gamma) \tag{20}$$

$$\mathbf{s}_t = \gamma_t \mathbf{z}_L + (1-\gamma_t)\mathbf{v}_t \tag{21}$$

$$\mathbf{x}_t = \begin{pmatrix} \mathbf{E}\mathbf{y}_{t-1} \\ \mathbf{s}_t \end{pmatrix} \tag{22}$$

$\mathbf{W}_{\gamma y} \in \mathbb{R}^{1\times m}, \mathbf{W}_{\gamma h} \in \mathbb{R}^{1\times n}$, and b_γ are learned projection matrices and bias randomly initialized. Constructing the fusion gate requires the number of dimensions of $\mathbf{z}_L$ and $\mathbf{v}_t$ to be equal. Thus, we make sure the rating prediction component outputs $\mathbf{z}_L \in \mathbb{R}^D$.

The output probabilities are computed based on previous word, the LSTM state, and the context vector from fusion gate:

$$\mathbf{g}_t = \mathbf{W}_g\phi(\mathbf{W}_{gy}\mathbf{E}\mathbf{y}_{t-1} + \mathbf{W}_{gh}\mathbf{h}_t + \mathbf{W}_{gs}\mathbf{s}_t) \tag{23}$$

$$p(\mathbf{y}_{tj}|\mathbf{x}_t) = \frac{\exp(\mathbf{g}_{tj})}{\sum_{k=1}^{V}\exp(\mathbf{g}_{tk})} \tag{24}$$

$\mathbf{W}_g \in \mathbb{R}^{V\times m}, \mathbf{W}_{gy} \in \mathbb{R}^{m\times m}, \mathbf{W}_{gh} \in \mathbb{R}^{m\times n}, \mathbf{W}_{gs} \in \mathbb{R}^{m\times D}$ are learned projection matrices randomly initialized. The word with the highest output probability will be chosen as the generated one.

$$\mathbf{y}_t^* = \underset{\mathbf{y}_j, j\in V}{\text{argmax}}\, p(\mathbf{y}_{tj}|\mathbf{x}_t) \tag{25}$$

Discussion. The relative generality of *MRG* can be seen from how the full-fledged model could be reduced into simplified models that work with less information. As earlier stated, the model is still capable of generating rating prediction and synthesizing review text without visual features. Without rating prediction, the model is effectively solving a personalized image captioning problem where the generated text will describe the image based on the user's experience of the item. If both visual features and rating prediction

Algorithm 1 Parameter Learning with Stochastic Gradient Descent

Input: $\mathcal{U}, \mathcal{I}, \mathcal{X}^r, \mathcal{X}^s$, learning rate η
Output: learned model parameters $\{\mathbf{P}, \mathbf{Q}, \mathbf{E}, \Theta_r, \Theta_s\}$

initialization
  $\mathbf{P}, \mathbf{Q}, \Theta_r, \Theta_s \leftarrow$ randomly initialized
  $\mathbf{E} \leftarrow$ randomly initialized / pre-trained embeddings
while not converged **do**
  for all $(u, i, r) \in \mathcal{X}^r$ **do**
    $\mathbf{P}_u = \mathbf{P}_u - \eta \cdot \frac{\partial}{\partial \mathbf{P}_u} \mathcal{L}^r(u, i, r)$
    $\mathbf{Q}_i = \mathbf{Q}_i - \eta \cdot \frac{\partial}{\partial \mathbf{Q}_i} \mathcal{L}^r(u, i, r)$
    $\Theta_r = \Theta_r - \eta \cdot \frac{\partial}{\partial \Theta_r} \mathcal{L}^r(u, i, r)$
    for all $(j, k, d, m) \in \mathcal{X}^s$ **where** $(j == u)$ & $(k == i)$ **do**
      $\mathbf{P}_j = \mathbf{P}_j - \eta \cdot \frac{\partial}{\partial \mathbf{P}_j} \mathcal{L}^s(j, k, d, m)$
      $\mathbf{Q}_k = \mathbf{Q}_k - \eta \cdot \frac{\partial}{\partial \mathbf{Q}_k} \mathcal{L}^s(j, k, d, m)$
      $\Theta_s = \Theta_s - \eta \cdot \frac{\partial}{\partial \Theta_s} \mathcal{L}^s(j, k, d, m)$
      $\mathbf{E}_{w \in d} = \mathbf{E}_{w \in d} - \eta \cdot \frac{\partial}{\partial \mathbf{E}_{w \in d}} \mathcal{L}^s(j, k, d, m)$
    end for
  end for
end while
return $\{\mathbf{P}, \mathbf{Q}, \mathbf{E}, \Theta_r, \Theta_s\}$

are removed, the model is solving the problem of personalized review text generation without any further information. If the goal is solely to predict ratings, both visual features and review text can be removed, leaving the model with only rating prediction. In Section 4, we will investigate the respective contributions of various components towards the final model as well as the performance of different architectural variants in an ablation analysis.

3.2 Model Learning and Inference

Learning. Our model consists of two supervised learnable components. To learn the parameters based on rating information, the model minimizes the regularized square error on the set $\mathcal{X}^r$:

$$\mathcal{L}^r = \frac{1}{2} \sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{I}} \mathbb{1}_{u,i} (r_{u,i} - \hat{r}_{u,i})^2 + \frac{\lambda_{wr}}{2} \sum_{\theta \in \Theta_r} \|\theta\|_2^2 \|\theta\|_2^2 + \frac{\lambda_{ur}}{2} \sum_{u \in \mathcal{U}} \|\mathbf{P}_u\|_F^2 + \frac{\lambda_{ir}}{2} \sum_{i \in \mathcal{I}} \|\mathbf{Q}_i\|_F^2$$

$\mathbb{1}_{u,i}$ is the indicator function which is equal to 1 if user u rated item i and equal to 0 otherwise, $\|.\|$ denotes the Frobenius norm of matrix, Θ_r is the set of parameters of rating prediction component, and $\lambda_{wr}, \lambda_{ur}, \lambda_{ir}$ are hyper-paramenters for regularization.

To learn the review text generator, we update the parameters by minimizing the regularized negative log-likehood on the set $\mathcal{X}^s$:

$$\mathcal{L}^s = \sum_{(u,i,d,m) \in \mathcal{X}^s} \sum_{w \in d} -\log p(\mathbf{y}_w | \mathbf{x}) + \frac{\lambda_{ws}}{2} \sum_{\theta \in \Theta_s} \|\theta\|_2^2 + \frac{\lambda_{us}}{2} \sum_{u \in \mathcal{U}} \|\mathbf{P}_u\|_F^2 + \frac{\lambda_{is}}{2} \sum_{i \in \mathcal{I}} \|\mathbf{Q}_i\|_F^2$$

where Θ_s is the set of parameters of rating prediction component, and $\lambda_{ws}, \lambda_{us}, \lambda_{is}$ are hyper-paramenters for regularization.

Algorithm 2 Multimodal Review Generation with Greedy Search

Input: user (u), item (i), image (m)
Output: estimated rating ($\hat{r}$), generated review text ($\hat{d}$)

initialization
  $\mathbf{P}_u, \mathbf{Q}_i \leftarrow$ user and item embeddings
  $t = 0$, $\mathbf{T} \leftarrow$ maximum review length
  $\Pi = [] \leftarrow$ generated word array
  $w_0 =$ "<start>", $\mathbf{y}_0 = \mathbf{y}_{w_0} \leftarrow$ starting word
  $\mathbf{c}_0, \mathbf{h}_0 \leftarrow$ LSTM initialized states from $[\mathbf{P}_u; \mathbf{Q}_i]$
  $\mathcal{A}_m = \text{CNN}(m) \leftarrow$ generated annotations from image m
$\mathbf{z}_L \leftarrow$ generated sentiment features from $[\mathbf{P}_u; \mathbf{Q}_i]$
$\hat{r} = \mathbf{W}_r \mathbf{z}_L + b_r$
while $t < T$ **do**
  $t = t + 1$
  $\mathbf{E}\mathbf{y}_{t-1} \leftarrow$ input word embedding
  $\mathbf{v}_t \leftarrow$ generated visual context from $\mathcal{A}_m$
  $\mathbf{s}_t \leftarrow$ generated fusion context from $\mathbf{z}_L$ and $\mathbf{v}_t$
  $\mathbf{x}_t = [\mathbf{E}\mathbf{y}_{t-1}; \mathbf{s}_t] \leftarrow$ input of the LSTM
  $\mathbf{y}_t = \underset{\mathbf{y}_j, j \in V}{\text{argmax}}\, p(\mathbf{y}_{tj} | \mathbf{x}_t) \leftarrow$ greedy selection
  $w_t = \text{map\_to\_word}(\mathbf{y}_t)$
  if $w_t ==$ "<end>" **then**
    break
  end if
  Π.append(w_t)
end while
$\hat{d} = \text{join}(\Pi)$
return $\{\hat{r}, \hat{d}\}$

The final objective of the model is the weighted combination of $\mathcal{L}^r$ and $\mathcal{L}^s$:

$$\mathcal{J} = \underset{\mathbf{P}, \mathbf{Q}, \mathbf{E}, \Theta_r, \Theta_s}{\text{minimize}} \Big(\frac{\lambda_r}{2} \sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{I}} \mathbb{1}_{u,i} (r_{u,i} - \hat{r}_{u,i})^2 + \lambda_s \sum_{(u,i,d,m) \in \mathcal{X}^s} \sum_{w \in d} -\log p(\mathbf{y}_w | \mathbf{x}) + \frac{\lambda_{wr}}{2} \sum_{\theta \in \Theta_r} \|\theta\|_2^2 + \frac{\lambda_{ws}}{2} \sum_{\theta \in \Theta_s} \|\theta\|_2^2 + \frac{\lambda_u}{2} \sum_{u \in \mathcal{U}} \|\mathbf{P}_u\|_F^2 + \frac{\lambda_i}{2} \sum_{i \in \mathcal{I}} \|\mathbf{Q}_i\|_F^2 \Big)$$

λ_r, λ_s are hyper-parameters to balance the importance between two components, $\lambda_{wr}, \lambda_{ws}, \lambda_u, \lambda_i$ are hyper-parameters for regularization of Θ_r, Θ_s, user embeddings, item embeddings respectively.

We derive a learning algorithm based on stochastic gradient descent using backpropagation (Alg. 1). Each triple (u, i, r) in $\mathcal{X}^r$ may be associated with quadruples (u, i, d, m) in $\mathcal{X}^s$. For each observation in $\mathcal{X}^r$, we update user, item embeddings and parameters in Θ_r by minimizing $\mathcal{L}^r$. With each $(u, i, r) \in \mathcal{X}^r$, we find corresponding $(u, i, d, m) \in \mathcal{X}^s$, and update user, item embeddings, parameters in Θ_s, word embeddings by minimizing $\mathcal{L}^s$. One might argue that the model can be updated based on observations from $\mathcal{X}^s$ before $\mathcal{X}^r$. We update rating prediction first because review text generation requires features from the rating, supplying the LSTM with more accurate sentiment signals. We can speed up the training with mini-batch gradient descent yielding a faster rate of convergence.

Table 2: Data Statistics

Data	#users	#items	#ratings	#docs	#images
CH	2,908	2,725	19,453	82,283	38,978
LA	36,918	23,601	355,553	1,539,355	680,892
NY	21,474	15,160	199,723	818,682	382,368
SF	7,999	3,375	65,228	320,165	139,014

Inference. To generate a review given a user u, an item i and an image m, we derive an inference algorithm for our *MRG* model (see Alg. 2). The rating $\hat{r}_{u,i}$ is first estimated using rating prediction component that also outputs the sentiment features $\mathbf{z}_L$. The review text generation is based on greedy strategy, where at each time step t, the word with the highest probability is chosen for the review.

4 EXPERIMENTS

Our experimental objective is to investigate the following research questions with respect to the proposed model *MRG*.

- **RQ#1:** Does factoring content via the review generation task improve the performance of *MRG* in rating prediction compared to rating-only matrix factorization approaches?
- **RQ#2:** Does coupling rating prediction and text generation at the review level help *MRG* to outperform comparable content-based recommender systems in rating prediction?
- **RQ#3:** Does factoring sentiment and visual features help *MRG* to generate review texts closer to the ground truth?
- **RQ#4:** How do the different components of *MRG* contribute to its performance? How do different architecture variants perform in rating prediction and review text generation?

4.1 Experimental Setup

Datasets. We use datasets of online reviews crawled from *Yelp* covering 4 US cities, namely: Chicago (CH), Los Angeles (LA), New York (NY), and San Francisco (SF). Table 2 shows some statistics about these datasets. Within each online review, there is a rating, a review text, and one or more images taken by the user. The set $\mathcal{X}^r$ consists of all rating observations. The set $\mathcal{X}^s$ is constructed as follows. First, we split the review text into shorter passages; in this case, each sentence is considered a review document d. A review image m could be paired with multiple documents. To identify the best-matching documents to an image, we rank sentences within the same review based on the cosine similarity of their TF-IDF scores to that of the user-provided image description, and consider up to 5 highest-ranked documents above a threshold (0.1) to be relevant. These would then form the (u, i, d, m) quadruples in $\mathcal{X}^s$.

Task #1: Rating Prediction. One evaluation task is rating prediction. Since *Yelp* uses rating scores $r_{u,i}$ from 1 to 5, our model learns to estimate ratings within that range. We rely on two standard metrics for rating prediction, namely: Mean Absolute Error (*MAE*) and Root Mean Square Error (*RMSE*) as shown below.

$$MAE = \frac{1}{N} \sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{I}} \mathbb{1}_{u,i} |r_{u,i} - \hat{r}_{u,i}|$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{I}} \mathbb{1}_{u,i} (r_{u,i} - \hat{r}_{u,i})^2}$$

Table 3: Hyper-parameter settings

Method	Hyper-parameters
PMF	latent factors = 10, $\lambda_U = \lambda_V = 0.1$
NMF	latent factors = 30
SVD++	latent factors = 10, $\lambda = 0.01$
CDL	$K = 50, \lambda_v = 10, \lambda_n = 1000, \lambda_u = \lambda_w = 0.1$
ConvMF	$k = 50, p = 200, \lambda_u = 1, \lambda_v = 100$
DeepCoNN	$\lvert x_u \rvert = \lvert y_i \rvert = 50, c = 200, t = 3$
Att2Seq	$m = 200, r = 200, n = 256, L = 2$
SAT	$m = 200, n = 256, \lambda = 1$
MRG	$K = 200, m = 200, n = 256, L = 2, \lambda_r = \lambda_s = 1.0, \lambda_{wr} = \lambda_{ws} = \lambda_u = \lambda_i = 0.0001$

$\mathbb{1}_{u,i}$ is the indicator function which is equal to 1 if user u rated item i in the testing set and equal to 0 otherwise, N indicates the total number of rating observations in the testing set.

For RQ#1 to validate the use of content, we compare *MRG* with the following rating-based matrix factorization methods.

- **PMF**: Probabilistic Matrix Factorization [22] models matrix factorization under probabilistic framework with the assumptions of Gaussian distribution for rating values.
- **NMF**: Non-negative Matrix Factorization [15] ensures that the factorized user and item matrices are non-negative.
- **SVD++**: Singular Value Decomposition [12] learns from ratings as well as considers neighborhood information.

For RQ#2 to validate the modeling of content at review level, we compare *MRG* with the following content-based recommendation methods that incorporate textual content at the item level.

- **CDL**: Collaborative Deep Learning [33] uses Stacked Denoising AutoEncoder (SDAE) to learn item features.
- **ConvMF**: Convolutional Matrix Factorization [10] replaces the SDAE with a convolutional neural network (CNN).
- **DeepCoNN**: [38] models both user and item features from aggregated review texts using siamese CNN.

Task #2: Review Text Generation. The other evaluation task is generating review text. BLEU [26] is a well-known metric for evaluating the quality of generated text, which has been widely used for machine translation and image captioning. We use smoothed BLEU [18] and report the results of BLEU scores from 1 to 4. Another metric is ROUGE [17] which has been extensively used for text summarization. We report the F-measure, which is the geometric mean of the precision and recall of ROUGE-1 (covering 1-grams) and ROUGE-L (covering the longest subsequences) respectively.

A review may be associated with multiple photos. We therefore evaluate the generated text under two scenarios: photo level and rating level. Photo level means the generated text for each photo is evaluated individually, and we report the average results over all the photos. Rating level will first aggregate (i.e., average) the results associated with all the photos of the same review, and then we report the average results across all the ratings. For the latter scenario, the ground-truth references corresponding to the photos of the same rating are merged. The purpose is to make the comparison fair for the baselines not using visual information.

Table 4: Review text generation performance evaluated @ photo level (*higher is better*)

	Metric	Att2Seq		SAT	MRG
		-Rating	*+Rating*		
Chicago	**BLEU-1**	29.07	29.49	29.89	**33.34**
	BLEU-2	15.57	15.79	15.46	**17.28**
	BLEU-3	11.87	11.88	11.75	**12.53**
	BLEU-4	10.51	10.41	10.42	**10.77**
	ROUGE-1	22.55	22.19	24.13	**25.17**
	ROUGE-L	17.21	16.72	17.77	**18.43**
Los Angeles	**BLEU-1**	30.11	30.67	31.63	**32.37**
	BLEU-2	16.95	17.22	17.20	**17.98**
	BLEU-3	12.98	13.15	13.16	**13.64**
	BLEU-4	11.40	11.55	11.57	**12.01**
	ROUGE-1	21.86	22.99	23.38	**24.22**
	ROUGE-L	16.65	17.71	17.96	**18.50**
New York	**BLEU-1**	28.95	29.40	29.68	**30.67**
	BLEU-2	16.24	15.75	16.09	**16.92**
	BLEU-3	12.46	11.85	12.30	**12.86**
	BLEU-4	11.00	10.26	10.82	**11.28**
	ROUGE-1	22.25	22.11	23.46	**24.09**
	ROUGE-L	16.96	16.65	17.83	**18.35**
San Francisco	**BLEU-1**	30.19	30.74	30.64	**32.24**
	BLEU-2	16.86	17.15	16.66	**17.76**
	BLEU-3	12.90	12.94	12.55	**13.31**
	BLEU-4	11.34	11.25	10.90	**11.60**
	ROUGE-1	22.17	21.83	22.25	**25.20**
	ROUGE-L	16.77	16.63	16.91	**18.39**

Table 5: Review text generation performance evaluated @ rating level (*higher is better*)

	Metric	Att2Seq		SAT	MRG
		-Rating	*+Rating*		
Chicago	**BLEU-1**	36.84	37.12	37.75	**42.00**
	BLEU-2	19.60	19.74	19.30	**21.74**
	BLEU-3	14.47	14.35	14.19	**15.16**
	BLEU-4	12.52	12.27	12.31	**13.04**
	ROUGE-1	22.41	21.96	24.03	**24.93**
	ROUGE-L	17.11	16.55	17.64	**18.18**
Los Angeles	**BLEU-1**	36.86	37.76	38.26	**39.38**
	BLEU-2	20.72	21.21	20.58	**21.94**
	BLEU-3	15.47	15.79	15.38	**16.30**
	BLEU-4	13.32	13.58	13.29	**14.07**
	ROUGE-1	21.61	22.89	22.66	**23.74**
	ROUGE-L	16.41	17.52	17.34	**18.08**
New York	**BLEU-1**	35.94	36.09	36.37	**38.03**
	BLEU-2	20.18	19.33	19.53	**20.97**
	BLEU-3	15.11	14.15	14.58	**15.51**
	BLEU-4	13.08	12.00	12.62	**13.32**
	ROUGE-1	21.88	21.96	22.84	**23.79**
	ROUGE-L	16.63	16.55	17.33	**18.11**
San Francisco	**BLEU-1**	39.55	39.83	38.78	**41.51**
	BLEU-2	22.10	22.25	20.81	**22.96**
	BLEU-3	16.39	16.26	15.20	**16.75**
	BLEU-4	14.03	13.76	12.93	**14.29**
	ROUGE-1	22.19	22.12	21.86	**25.54**
	ROUGE-L	16.76	16.85	16.62	**18.55**

For RQ#3, to validate review text generation, we compare with two text generation methods constructing three baselines.

- **Att2Seq**: Attribute-to-Sequence [5] treats user, item, and rating as three attributes and generates review text with the attention-based multilayer LSTMs. We compare with two versions of Att2Seq which are: without observing rating (*-Rating*) that is similar to our setting and with observed rating (*+Rating*) that is the authors' original setting (which confers it an advantage since the ground-truth rating is given).
- **SAT**: Show, Attend, and Tell [35] is the state-of-the-art method for image captioning, which exploits visual features using attention mechanism and generates the captions using LSTM.

Experimental Settings. For each dataset in Table 2, we reserve 10% of the data for model validation. The remaining 90% is split into training and testing sets. For evaluating rating prediction, we report results for different training percentages (80%, 60%, 40%, and 20%). This is to examine the effects of sparsity on rating prediction, which is a crucial issue for recommender systems. For evaluating review text generation, we use the same split, yet we report the results for 80% training and 20% testing due to space limitations.

We tune all the comparative models for their respective best performance. Table 3 indicates the discovered hyper-parameter settings used for different methods. For PMF, NMF, and SVD++, we use grid search to find the number of latent factors from $\{10, \ldots, 100\}$ where we end up with 10 for PMF, 30 for NMF, and 10 for SVD++. The regularization hyper-parameters are searched from $\{0.001, 0.01, 0.1, 1.0\}$.

For CDL, ConvMF, and DeepCoNN, the number of item features is grid-searched from $\{10, 50, 100, 200\}$ and 50 gives the best performance and fair comparison between three methods, whereas $\{100, 200\}$ cause overfitting. Regularization hyper-parameters of the models are tuned using the validation set.

For LSTM-based models Att2Seq, SAT, and *MRG*, we use the same number of dimensions of the LSTM cells (256). SAT and *MRG* use one-layer LSTM whereas Att2Seq uses two layers of LSTMs as we respect the original design of the authors. In practice, we observe negligible improvements of using more than one layer for *MRG* so we keep using one layer for the gain in efficiency. Att2Seq and *MRG* use the same number of dimensions for user and item embeddings as well as rating embeddings for the former. Other hyper-parameters are tuned using the validation set. In testing, the maximum length of generated text is set at $T = 20$ for all the methods. *MRG* [1] is implemented using Tensorflow [1] and trained with batch size of 64. We update the parameters using Adam [11] rule with learning rate $\eta = 0.0003$, $\beta_1 = 0.9$, $\beta_2 = 0.99$, and $\epsilon = 10^{-8}$.

The vocabulary is built from words appearing at least 3 times in training and validation sets. Infrequent words are replaced with special <UNK> token. All models using word embeddings are initialized from Glove [27] pre-trained word embeddings of 200 dimensions.

[1] https://code.preferred.ai/mrg

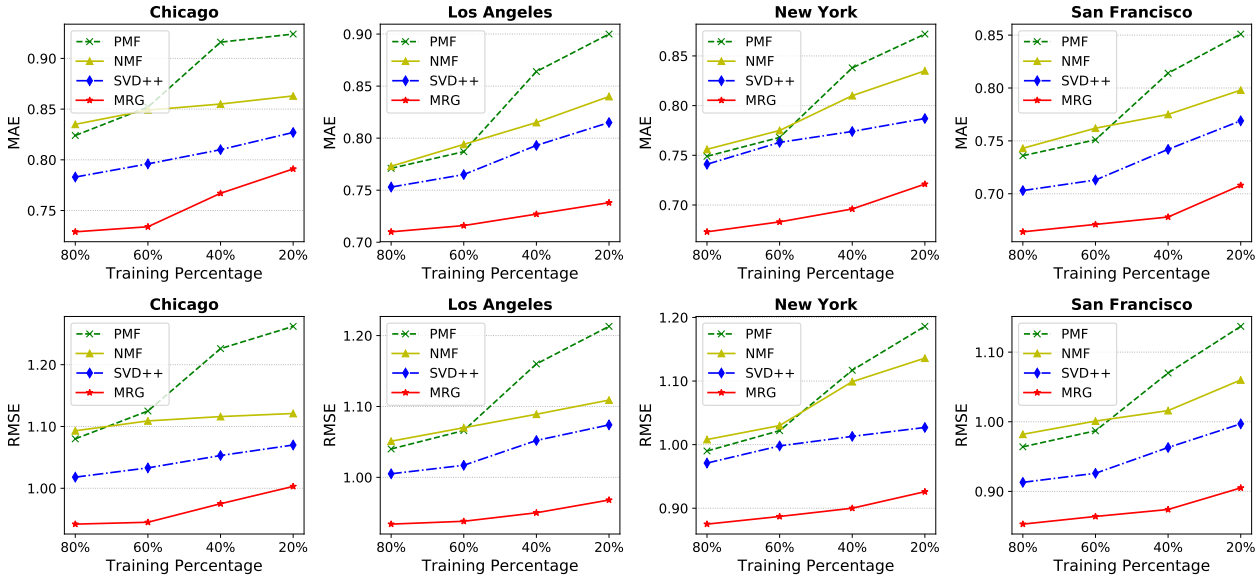


Figure 4: Rating prediction comparision with content-based approaches (*lower is better*) (The results are statistically significant with $p < 0.01$ based on the paired sample t-test)

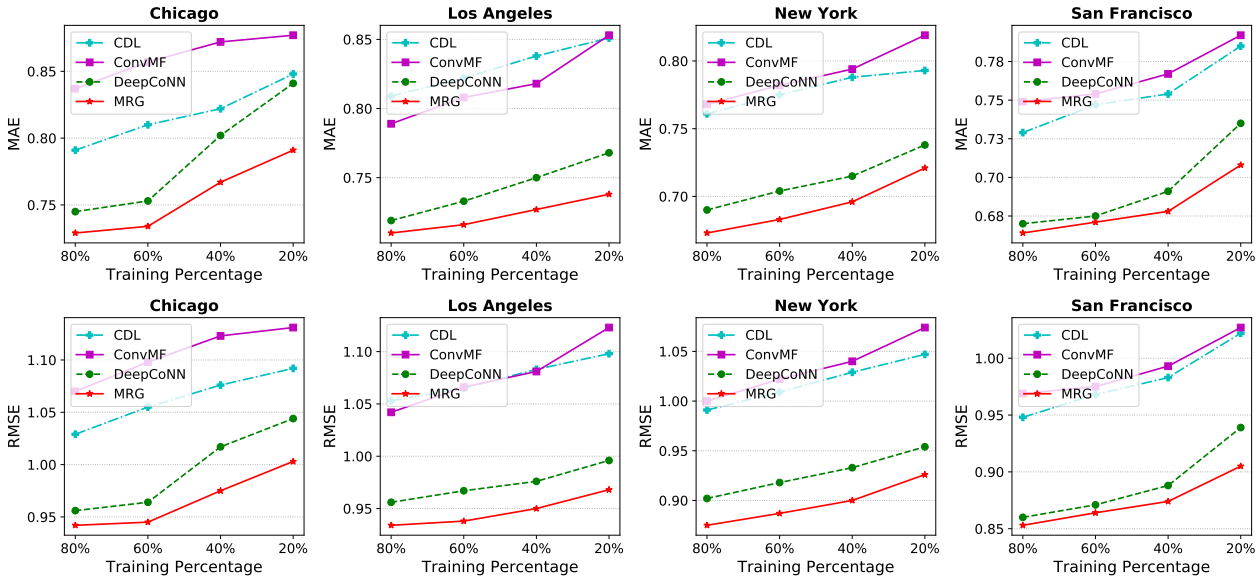


Figure 5: Rating prediction comparision with content-based approaches (*lower is better*) (The results are statistically significant with $p < 0.01$ based on the paired sample t-test)

4.2 Comparison to Baselines (RQ#1 to RQ#3)

We now compare the relative performance of *MRG* with respect to the baselines, corresponding to the first three research questions.

RQ#1. First, we consider the effects of using content for rating prediction. Figure 4 shows the performance of *MRG* in rating prediction compared to matrix factorization approaches as we vary the degree of sparsity. Lower MAE or RMSE implies higher accuracy. Expectedly, with less training data, the performance generally worsens because greater sparsity may cause overfitting.

In relative terms, the clear consensus across four datasets and two metrics is that *MRG* outperforms the baselines. This validates the advantage of modeling content in addition to ratings.

Among the three baselines, SVD++ with the advantage of modeling neighborhood information tends to perform better than PMF and NMF. In turn, PMF performs slightly better than NMF with 80% data in training, but deteriorates quickly with more sparsity.

RQ#2. We now compare *MRG* with content-based recommendation methods, namely: CDL, ConvMF, and DeepCoNN. Figure 5 illustrates the performance of these models. Here, it is also evident that *MRG* outperforms the baselines, which we attribute to its modeling content at the review level. In contrast, CDL and ConvMF that model content at item level (aggregating the reviews of an item into one document) tend to perform similarly across the datasets, with the exception of Chicago where CDL seems to have an edge. In turn, DeepCoNN performs better than both CDL and ConvMF, because DeepCoNN exploits review texts for both item and user.

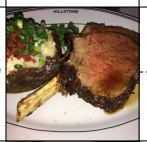
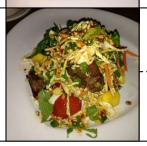
RQ#3. Table 4 and Table 5 show the results of review text generation at photo level and rating level respectively with 80% training. The baselines are text generation methods Att2Seq that relies on user, item, rating, and SAT that relies on image. For all the datasets, *MRG* consistently has higher BLEU and ROUGE scores than the baselines, indicating that it synthesizes review texts that are closer to the ground-truth references. This helps to validate the joint modeling of rating prediction and review text generation.

Examining the baselines, we see that for Chicago, Los Angeles, and San Francisco, Att2Seq with rating (*+Rating*) performs slightly better than without rating (*-Rating*). That indicates that sentiment information encoded in the rating does contribute to review text generation. In some cases, SAT that uses image information is slightly better than Att2Seq (no image). However, this is an imperfect comparison as Att2Seq factors in user and item, whereas SAT does not. For a clearer sense, we conduct an ablation analysis.

4.3 Ablation Analysis (RQ#4)

To investigate the respective contribution of our architecture components to the overall performance of *MRG*, we conduct an ablation analysis. Table 6 shows the rating prediction performance of our model when we systematically remove different components as discussed at the end of Section 3.1. The ticks indicate which information is included. The model with rating prediction alone (*User + Item*) is the worst, performing at similar levels as the matrix factorization baselines. The review text generation component (*Text*) sharply boosts the results because of the strongly evident preference information from text. When the visual features (*Photo*) are introduced, the sentiment features are aligned more accurately with the sentiment words during the text generation process, which in turn benefits the learned representations of the rating prediction. The improvements are even clearer as the data becomes sparser.

Table 7 and Table 8 show the review text generation quality at the photo level and the rating level respectively. The model with only (*User + Item*) performs the worst, as the generated text becomes generic because of the loss of the guiding information during generation process. Adding either the sentiment signal (*Rating*) or the visual signal (*Photo*) individually improves the results, indicating that these components are useful. The gain of adding *Rating* is relatively higher than adding *Photo*. The final model that combines both signals consistently outperforms the rest, demonstrating the utility of joining these components in the overall architecture. Due to space limitation, we omit BLEU-2 and BLEU-3 measurements as both of them show the same trend as discussed.

	Photo	Rating	Review
Image 1		**4.1**	**the steak was cooked perfectly .**
		4.0	order the medium rare my favorite and you will have yourself a big , fat , and juicy steak to shove between your big smile .
Image 2		**4.1**	**the beef was cooked perfectly .**
		4.0	i 'll recommend the thai steak and noodle salad to you .

(a) Case Study #1: A user (Ronald "Exotic food consumer" L.) reviews an item (Hillstone) with different images.

	Photo	Rating	Review
Ellen "FuZe" Z.		**4.5**	**the clam chowder was good .**
		5.0	best clam chowder i 've ever had .
Young Y.		**3.4**	**the clam chowder was a bit too salty .**
		3.0	the boston clam chowder was pretty salty and i 've had lots of clam chowder before .

(b) Case Study #2: An item (Tadich Grill) is reviewed by two users with different sentiments.

	Photo	Rating	Review
Philz Coffee		**4.6**	**i 've had a few times for the best breakfast sandwich .**
		4.0	the avocado toast was surprisingly good .
A16		**4.2**	**i was n't sure to try the pizza .**
		3.0	i think i might want to try their other pizzas which might be better tasting than their funghi .

(c) Case Study #3: A user (Rodney "Hungry Trikker" H.) reviews two different items.

Figure 6: Multimodal review generation. The first line next to each photo (bold) is generated rating & text, and the second line is the ground truth. Photos are best seen in color.

4.4 Case Studies

To gain an intuitive sense of the workings of the *MRG* model, we illustrate several examples of the rating prediction as well as the synthesized text generated by *MRG*. The first case study in Fig. 6a examines the text generated for two photos by the same user for an item. The predicted rating (bold) is close to the ground-truth and is the same for both photos since it is for the same item. The predicted texts (bold) also have some similarity to the ground-truth. Interestingly, the texts corresponding to the photos are different

Table 6: Ablation analysis: rating prediction performance (*lower is better*)
(The results are statistically significant with $p < 0.01$ based on the paired sample t-test)

	Dataset	Chicago			Los Angeles			New York			San Francisco		
	User + Item	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Text		✓	✓		✓	✓		✓	✓		✓	✓
	Photo			✓			✓			✓			✓
MAE	Training 80%	0.751	0.741	**0.729**	0.749	0.713	**0.710**	0.695	0.677	**0.673**	0.685	0.667	**0.664**
	Training 60%	0.779	0.751	**0.734**	0.766	0.719	**0.716**	0.717	0.693	**0.683**	0.703	0.678	**0.671**
	Training 40%	0.832	0.788	**0.767**	0.801	0.732	**0.727**	0.754	0.704	**0.696**	0.743	0.686	**0.678**
	Training 20%	0.920	0.817	**0.791**	0.881	0.746	**0.738**	0.820	0.740	**0.721**	0.811	0.728	**0.708**
RMSE	Training 80%	1.006	0.952	**0.942**	0.985	0.936	**0.934**	0.912	0.881	**0.875**	0.889	0.857	**0.853**
	Training 60%	1.032	0.961	**0.945**	1.004	0.943	**0.938**	0.937	0.900	**0.887**	0.941	0.872	**0.864**
	Training 40%	1.083	1.010	**0.975**	1.043	0.957	**0.950**	0.981	0.910	**0.900**	0.989	0.879	**0.874**
	Training 20%	1.165	1.039	**1.003**	1.126	0.988	**0.968**	1.082	0.953	**0.926**	1.058	0.932	**0.905**

Table 7: Ablation analysis: review text generation performance evaluated @ photo level (*higher is better*)

	User + Item	✓	✓	✓	✓
	Rating		✓		✓
	Photo			✓	✓
CH	**BLEU-1**	29.98	32.16	32.19	**33.34**
	BLEU-4	9.37	10.32	9.83	**10.77**
	ROUGE-1	23.87	23.82	24.05	**25.17**
	ROUGE-L	17.29	17.72	17.61	**18.43**
LA	**BLEU-1**	30.37	32.04	31.41	**32.37**
	BLEU-4	11.23	11.57	11.22	**12.01**
	ROUGE-1	22.07	23.41	23.17	**24.22**
	ROUGE-L	17.19	17.97	17.76	**18.50**
NY	**BLEU-1**	29.23	29.95	29.45	**30.67**
	BLEU-4	10.68	10.98	10.90	**11.28**
	ROUGE-1	22.77	23.67	23.10	**24.09**
	ROUGE-L	17.59	18.23	17.73	**18.35**
SF	**BLEU-1**	29.64	31.56	31.09	**32.24**
	BLEU-4	10.86	11.19	10.61	**11.60**
	ROUGE-1	21.34	22.74	22.08	**25.20**
	ROUGE-L	16.49	17.30	16.89	**18.39**

Table 8: Ablation analysis: review text generation performance evaluated @ rating level (*higher is better*)

	User + Item	✓	✓	✓	✓
	Rating		✓		✓
	Photo			✓	✓
CH	**BLEU-1**	36.64	40.42	39.63	**42.00**
	BLEU-4	10.65	12.04	11.22	**13.04**
	ROUGE-1	23.76	23.83	23.34	**24.93**
	ROUGE-L	17.10	17.52	17.06	**18.18**
LA	**BLEU-1**	37.08	39.18	37.97	**39.38**
	BLEU-4	13.01	13.45	12.85	**14.07**
	ROUGE-1	21.89	23.21	22.59	**23.74**
	ROUGE-L	17.00	17.73	17.32	**18.08**
NY	**BLEU-1**	35.87	37.04	36.19	**38.03**
	BLEU-4	12.44	12.95	12.85	**13.32**
	ROUGE-1	22.40	23.55	22.58	**23.79**
	ROUGE-L	17.32	18.09	17.38	**18.11**
SF	**BLEU-1**	38.51	40.67	39.50	**41.51**
	BLEU-4	13.19	13.64	12.63	**14.29**
	ROUGE-1	21.44	22.96	22.25	**25.54**
	ROUGE-L	16.58	17.52	17.06	**18.55**

in their objects ("steak" vs. "beef") rather than in their sentiment, signalling the utility of images in differentiating content in the text generation. The second case study in Fig. 6b examines two users with different sentiments for an item, as reflected by their ratings. Here, both photos depict clam chowder, but the generated texts are different: "good" for the more positive user, and "salty" for the more negative user. Finally, the third case study in Fig. 6c depicts how the same user may review two items. In one case, the object of interest is a toast or sandwich. In the other, it is pizza. These are examples of how rating prediction interacts with review text generation sharing some sentiment signals, and how the visual features help to align with aspect words allowing the sentiment features to align with the sentiment words.

5 CONCLUSION

We propose *MRG*, a neural model for multimodal review generation that joins a rating prediction component and a review text generation component informed by visual features. Through comprehensive experiments and ablation analysis, we establish its utility for recommendation tasks, outperforming both content-based recommender systems and review text generation baselines. We attribute the outperformance to a more holistic representation of various signals within a review, yielding more accurate recommendations.

ACKNOWLEDGMENTS

This research is supported by the National Research Foundation, Prime Minister's Office, Singapore under its NRF Fellowship Programme (Award No. NRF-NRFF2016-07).

REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: a system for large-scale machine learning.. In *OSDI*, Vol. 16. 265–283.

[2] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. 2007. Multi-task feature learning. In *Advances in neural information processing systems*. 41–48.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. Ieee, 248–255.

[5] Li Dong, Shaohan Huang, Furu Wei, Mirella Lapata, Ming Zhou, and Ke Xu. 2017. Learning to generate product reviews from attributes. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, Vol. 1. 623–632.

[6] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*. IEEE, 6645–6649.

[7] Xiangnan He, Tao Chen, Min-Yen Kan, and Xiao Chen. 2015. Trirank: Review-aware explainable recommendation by modeling aspects. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. 1661–1670.

[8] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 173–182.

[9] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[10] Donghyun Kim, Chanyoung Park, Jinoh Oh, Sungyoung Lee, and Hwanjo Yu. 2016. Convolutional matrix factorization for document context-aware recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 233–240.

[11] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[12] Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 426–434.

[13] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 8 (2009), 30–37.

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. 1097–1105.

[15] Daniel D Lee and H Sebastian Seung. 2001. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*. 556–562.

[16] Piji Li, Zihao Wang, Zhaochun Ren, Lidong Bing, and Wai Lam. 2017. Neural rating regression with abstractive tips generation for recommendation. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 345–354.

[17] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out* (2004).

[18] Chin-Yew Lin and Franz Josef Och. 2004. Orange: a method for evaluating automatic evaluation metrics for machine translation. In *Proceedings of the 20th international conference on Computational Linguistics*. Association for Computational Linguistics, 501.

[19] Guang Ling, Michael R Lyu, and Irwin King. 2014. Ratings meet reviews, a combined approach to recommend. In *Proceedings of the 8th ACM Conference on Recommender systems*. ACM, 105–112.

[20] Julian McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*. ACM, 165–172.

[21] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[22] Andriy Mnih and Ruslan R Salakhutdinov. 2008. Probabilistic matrix factorization. In *Advances in neural information processing systems*. 1257–1264.

[23] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. 2011. Multimodal deep learning. In *Proceedings of the 28th international conference on machine learning (ICML-11)*. 689–696.

[24] Jianmo Ni, Zachary C Lipton, Sharad Vikram, and Julian McAuley. 2017. Estimating reactions and recommending products with generative models of reviews. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Vol. 1. 783–791.

[25] Jianmo Ni and Julian McAuley. 2018. Personalized Review Generation by Expanding Phrases and Attending on Aspect-Aware Representations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Vol. 2. 706–711.

[26] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, 311–318.

[27] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.

[28] Steffen Rendle. 2010. Factorization machines. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*. IEEE, 995–1000.

[29] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*. 452–461.

[30] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[31] Xiaoyuan Su and Taghi M Khoshgoftaar. 2009. A survey of collaborative filtering techniques. *Advances in artificial intelligence* 2009 (2009).

[32] Chong Wang and David M Blei. 2011. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 448–456.

[33] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1235–1244.

[34] Zhongqing Wang and Yue Zhang. 2017. Opinion recommendation using neural memory model. In *Proceedings of the 2017 conference on empirical methods in natural language processing (EMNLP)*. 1626–1637.

[35] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*. 2048–2057.

[36] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. 2016. Image captioning with semantic attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4651–4659.

[37] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. 2014. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. 83–92.

[38] Lei Zheng, Vahid Noroozi, and Philip S Yu. 2017. Joint deep modeling of users and items using reviews for recommendation. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. ACM, 425–434.