# Aspect-level Item Recommendation Based on User Reviews with Variational Autoencoders

Wei Ou[a], Van-Nam Huynh[b], Binyi Ou[c]

[a] *International Business School,*
*Zhejiang Gongshang University*
[b] *the School of Knowledge Science,*
*Japan Advanced Institute of Science and Technology*
[c] *YongZhou Normal College*

## Abstract

In this paper we propose an aspect-based recommendation model based on the variational autoencoder, that not only gives coarse predictions about what items may interest a user, but also the finer-grained predictions about what aspects the user may like about the recommended items. The proposed model first employs convolution operations to learn user representation distributions from user aspect-level sentiments on seen items, and then draw samples from the distributions and feed them to transpose convolution operations to 'reconstruct' the missing aspect-level sentiments for unseen items. To prevent overfitting, we impose a prior for the representation distributions and penalize the model if the learned distributions are distant from the prior. Based on the output of the proposed model, we propose a two-stage ranking scheme that combines aspect-level and overall sentiment signals to rank items. Experiment results show that the proposed model outperforms a number of state-of-the-art aspect-based recommendation models, and the two-stage ranking scheme improves the traditional ranking by the overall sentiment predictions.

*Keywords:* Aspect-based recommender systems, variational autoencoder, item ranking, aspect-level sentiment prediction

## 1. Introduction

Recommender systems, that aim to generate effective recommendations to help users quickly identify products or services interest them, have long been a very active topic in the community of information science. Over the past decades, the majority of the research efforts in this field focused on giving coarse predictions about the overall interest levels of users on unseen items, based on users' historical behavior on online platforms, e.g., click records, ratings and comments on items they have consumed. However, coarse predictions are usually difficult to interpret and offer little information to explain why a user may choose the recommended items. In recent years, aspect-based recommender systems (Bauman et al., 2017), that not only give overall predictions, but also finer-grained predictions on aspects of items, have quickly emerged as a popular topic in recommender system research literature.

Many of the existing aspect-based recommendation models introduce aspect-level latent factors (Koren et al., 2009) derived from user-item-aspect rating matrices (in this paper both 'sentiment' and 'rating' are indicators of user satisfaction level and therefore used interchangeably, though they take different value ranges) (Hernández-Rubio et al., 2019; Chambua & Niu, 2020), to represent the 'quality' of each aspect and the 'requirement' of users on the aspect. Aspect-level ratings of a user on an item are modeled as the result of the linear interactions among the latent factors. Though latent factor models are usually computationally effective and easy to implement, they suffer limited modeling capacity and are ineffective to capture the non-linearity in the interactions among the latent factors (Liang et al., 2018; Salakhutdinov & Mnih, 2008). In recent years, there also exist some research works that apply deep learning models to encode the interactions. However, since the user-item-aspect rating matrices are usually extremely sparse, they are highly prone to overfitting.

In this research, we aim to address the issues mentioned above. In other words, we aim to propose a neural aspect-based recommendation model that not only encodes the non-linearity in the interactions, but also enjoys lower risk of overfitting. We consider the variational autoencoder (VAE) a suitable model choice for the following reasons. First, it can employ very expressive model structures to capture the non-linearity. Second, as shown by Liang et al (Liang et al., 2018), the Bayesian approach of VAE provides strong regularization for the training process that effectively reduces the risk of overfitting. In this research our main research objective is to apply VAE to the aspect-based recommendation task, which breaks down into the following subtasks.

### *1.1. Research objectives and contributions*

- How to learn aspect-level representations of users and item, and how to encode the interactions among the representations to predict aspect-level sentiments, under the generative and probabilistic framework of VAE.

- The proposed model outputs both aspect-level and overall sentiment predictions for unseen items. A natural problem to be addressed is that how to combine aspect-level and overall sentiment predictions to rank items to generate recommendations. Existing aspect-based recommendation models usually rank items by overall sentiment predictions, and the role of aspect-level predictions in improving item ranking performance is rarely explored.

- Study the behavior of the proposed model and compare it against the existing latent factor based aspect-based recommendation models.

The main contributions of this work can be summarized as follows.

- We propose the Variational Autoencoder for Aspect-based Recommendation Model (VAEARM), in which we employ convolution and transpose convolution operations under a Bayesian structure to model the interactions between items and users. To the best of our knowledge, we are the first to employ VAE for aspect-based recommender systems.

- We propose a two-stage ranking scheme, as illustrated in Figure 1, that first ranks all candidate items by the overall predictions, and then reranks the top 100 items by combining the overall and aspect-level predictions. We observed obvious performance improvements of the ranking scheme on a range of linear and neural aspect-based recommendation models.

- Experiment results show that the proposed model outperforms a number of state-of-the-art baselines by significant margins. Analysis on the experiment results shows that, thanks to its Bayesian approach and expressive power, the proposed model is much less prone than the deterministic neural baselines to overfitting, and generates more informative aspect-level sentiment predictions than the linear baselines.

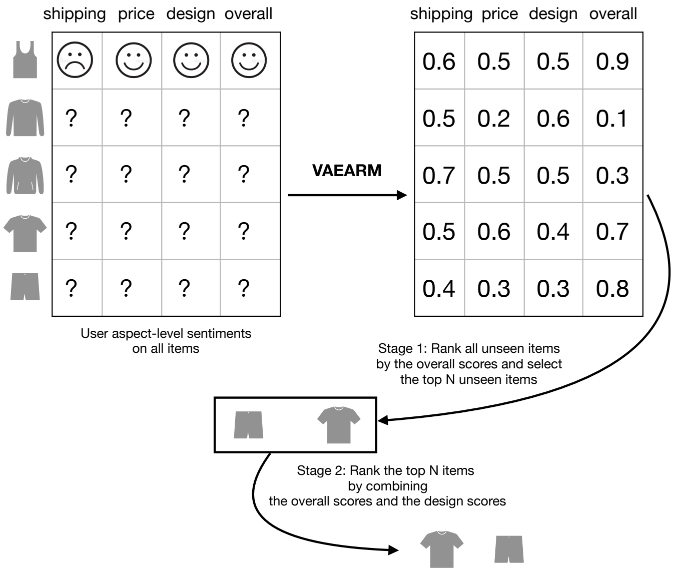

shipping price design overall
?
User aspect-level sentiments
on all items
VAEARM
shipping price design overall
0.6 0.5 0.5 0.9
0.5 0.2 0.6 0.1
0.7 0.5 0.5 0.3
0.5 0.6 0.4 0.7
0.4 0.3 0.3 0.8
Stage 1: Rank all unseen items
by the overall scores and select
the top N unseen items
Stage 2: Rank the top N items
by combining
the overall scores and the design scores

Figure 1: The workflow of the two-stage ranking scheme. The ranking scheme first ranks all candidate items by overall sentiment predictions, and then rerank the top items by combining overall and aspect-level sentiment predictions.

Also, to train the proposed model, it takes user aspect-level sentiments matrices, which are usually not readily available in many datasets. We use supervised sentence-level sentiment classifiers to detect the sentiments from user review text. However, training the classifiers requires labelled review sentences as the training data, which are very expensive to obtain. To avoid the labeling efforts, we treat each review as a single long sentence and the rating of the review as the sentiment label for the sentence. One problem with this approach is that sentence-level sentiment signal is lost in the training process. We adopt a sentence-level attention mechanism, inspired by the PARADE model (Li et al., 2020) which was originally proposed for information retrieval task, to maintain the sentence-level signal. This serves as a side contribution of this paper.

The rest of this paper is arranged as follows. Section 2 gives a brief introduction to related work. Section 3 and 4 introduce the proposed model in great detail. Section 5 introduces our approach to train sentence-level sentiment classifiers. Section 6 introduces the experimental setup and Section 7 presents the evaluation results. Section 8 gives a brief discussion about the finding from the experiment. Section 9 concludes this paper with theoretical and practical implications and future directions.

## 2. Related work

Compared with the abundant research works for general recommendation systems, the research works for aspect-based recommendation systems are rare. We divide existing aspect-based recommendation models into the following categories based on their working methodologies: explicit aspect-level profile based models and latent factors based models. Models of the first category usually explicitly build aspect-level profiles for users and items from various types of data, and generate recommendations based on the similarities of the profiles. Models of the second category usually introduce aspect-level latent factors for users and items, and model the aspect-level ratings as the results of the interactions among the latent factors. These models learn the latent factors from the training data and use their interactions on unseen items to generate recommendations.

### *2.1. Explicit aspect-level profile based models*

Dong et al. (Dong et al., 2013) extracted single nouns and bigram phrases from reviews to represent item aspects. Based on these terms, the authors identified sentiment orientations of each aspect to build product profiles. They authors compared search queries issued by shoppers against the product profiles to rank products based on a scoring mechanism. Liu et al. (Liu et al., 2013) first identified aspects and their sentiments from reviews to quantize the ‘concern’ and ‘requirement’ of each user, that represent the frequency of the user mentioning an aspect, and her requirement for the quality of the aspect, respectively. The authors score the relevance between a user and an item by the interactions among the ‘concern’ , ‘requirement’, and the average opinion on each aspect of the item. Chen et al. (Chen & Wang, 2013) first extracted user preferences over various item aspects, then clustered users by the preferences. The authors proposed to recommend a user the items that are highly rated by other users in the same cluster. Yang et al. (Yang et al., 2021) grouped reviews by their sentiment orientation, and extracted aspects and sentiments from the reviews to build sentiment-wise profiles for users and items. The authors proposed a voting mechanism that ranks items for a user by the combined similarities between the sentiment-wise profiles of each item and the user. Musto et al. (Musto et al., 2017) built item- and user-level profiles from aspects and sentiments extracted from reviews, and proposed a distance measure for the profile vectors to compare the similarities between users and items. In a realistic recommender system, the user-item-aspect rating matrices are extremely sparse, that causes high risk of overfitting to recommendation models. To alleviate the problem, Nilashi et al. (Nilashi et al., 2015) proposed to cluster users into homogeneous neighborhoods, and populate empty aspect-level rating entries associated with a user by the average of the ratings of other users from the same neighborhoods. Based on the processed user-item-aspect rating matrices, the authors aggregated the aspect-level ratings into overall ratings to generate recommendations based on a neural network. Instead of mining from reviews, Pan et al. (Pan et al., 2021) proposed to use tags of users to represent the aspect-level profiles of users and items. To overcome the data sparsity problem, the authors exploited the relations among tags and proposed a Bayesian network to expand the tag profiles.

### *2.2. Latent factor based models*

Diao et al. (Diao et al., 2014) proposed the JMARS model in which the authors employed three types of latent factor vectors: item- and user- and aspect-level latent factors, to model the aspect-level interactions between users and items. The authors assumed that the contribution of each aspect-level rating to the overall rating is determined by the mentioning probability of the aspect, which is further determined by another set of user- and item-wise parameters. At the meantime, the authors incorporated review text into the model and assumed it is generated from topic distributions whose parameters are dependent upon a set of aspect-specific parameters. The authors learned the model by maximizing the joint likelihood of overall ratings and review text. Wu et al. (Wu & Ester, 2015) proposed the FLAME model that shares a similar structure as the JMARS model, but employs finer-grained latent factors in order to better capture user personalized preferences. However, as described above, these models introduce too many dependency assumptions and do not generalize well across different datasets.

Chen et al. (Chen et al., 2016) proposed the LRPPM model that uses tensor factorization techniques (Frolov & Oseledets, 2017) to decompose aspect-level and overall sentiments into the pair-wise products of user-, item- and aspect-level latent factor matrices, and learn the matrices by optimizing a loss defined by the ranking of the user preferences over various aspects. Bauman et al. (Bauman et al., 2017) proposed the SULM model that introduces aspect-level latent vectors for each user and item, and assumes the interactions between the latent factors determine the aspect-level utility values of the item for the user. The aspect-level utility values are then combined into the overall utility value. The utility values are used to compute the aspect-level and overall sentiment distributions by the logistic function.

One common shortcoming of LRPPM and SULM is that they use linear functions to model the user-item-aspect interactions. The recent deep factorization models, represented by the CoSTCo model (Liu et al., 2019; Chen & Li, 2020) that employs convolutional neural networks to learn the latent factors, can partly address the linearity problem. However, since the training data is extremely sparse, the complex neural architectures of these models can easily cause overfitting. To alleviate the data sparsity problem, Chin et al. (Chin et al., 2018) proposed the ANR model that uses review text to learn the aspect-level latent factors and the interactions among them. Similarly, Liu et al. (Liu et al., 2020) proposed a deep multi-modal structure that learns user and item representations from review text and images, and generates recommendations with textual description of the aspects that may interest the target users. However, these models are extremely computational expensive and it is not practical to deploy them in large-scale recommender systems.

To deal with the data sparsity problem in general recommender systems, Liang et al. (Liang et al., 2018) proposed VAECF based on the variational autoencoder (VAE) and showed that the Bayesian approach of the model effectively reduces the risk of overfitting. Being inspired by VAECF, this paper also adopts the Bayesian approach and incorporates it into a highly expressive neural network structure for the aspect-based recommendation task. This is how our work differs from the existing work introduced above.

## 3. Proposed method

Suppose there is a set of $M$ items $I = \{i_m | m \in [1, M]\}$, and a set of users $U = \{u_n | n \in [1, N]\}$. Each of the items is evaluated by a set of $K$ aspects. Let $I_n \subseteq I$ denote the set of items a user $u_n$ has evaluated, $\hat{I}_n$ the unseen items for the user, $s_n = \{s_{nmk} | m \in [1, M], k \in [1, K]\}$ all the aspect-level sentiments of the user on all items. An aspect-level sentiment $s_{nmk}$ takes a binary value (either positive coded as 1, or non-positive coded as 0) if $i_m \in I_n$, otherwise no value. Our objective is to predict the missing values in $s_n$, namely the aspect-level sentiments of the user on unseen items based on the observed values.

### *3.1. VAEARM*

Given a user $u_n$, we assume her sentiments on the aspects of her evaluated items are generated by the following procedure:

- Draw a latent vector $z_n$ from $P(z)$
- For each item $i_m \in I_n$
    - For each aspect $k$
        - Generate an aspect-level sentiment $s_{nmk} \sim P(s_{nmk} | z_n)$

where $z$ is a vector with continuous values and can be viewed as the latent representation of the user. The generation process models the aspect-level interactions between the user and items.

Based on the assumption, the evidence likelihood $P(s_n)$ is formulated as follows:

$$\begin{aligned} P(s_n) &= \int_z P(z) P(s_n | z) dz &(1)\\ &= \int_z P(z) \prod_m \prod_k P(s_{nmk} | z)^{1\{i_m \in I_n\}} dz &(2) \end{aligned}$$

Where $1\{i_m \in I_n\}$ is an indicator function. It implies that we consider only the aspect-level sentiments on the seen items to compute the evidence likelihood. We are interested in deriving the posterior $P(z | s_n)$, namely, the distribution of the latent representations of the user given her aspect-level sentiments on seen items. With $z$ available, we can estimate the distributions of her aspect-level sentiments on unseen items. However, because of the presence of the integral over $z$ , $P(z|s_n)$ is intractable to compute. We use a simpler variational posterior $Q(z|s_n)$, to approximate $P(z|s_n)$. To make $Q(z|s_n)$ a good approximation, we need to minimize the Kullback–Leibler (KL) divergence between the two: $\mathcal{D}[Q(z|s_n)||P(z|s_n)]$. To minimize the KL divergence , we expand $\log P(s_n)$ as follows:

$$\begin{aligned}
\log P(s_n) =& \log \int_z P(z)P(s_n|z)dz && (3)\\
&= \log \int_z P(z)P(s_n|z)\frac{Q(z|s_n)}{Q(z|s_n)}dz && (4)\\
&= \log E_{z\sim Q(z|s_n)}\frac{P(z)P(s_n|z)}{Q(z|s_n)} && (5)\\
&= \underbrace{E_{z\sim Q(z|s_n)}[\log P(s_n|z)] - \mathcal{D}[Q(z|s_n)||P(z)]}_{ELBO} + \mathcal{D}[Q(z|s_n)||P(z|s_n)] && (6)
\end{aligned}$$

In the last equation, the term to the left of $\mathcal{D}[Q(z|s_n)||P(z|s_n)]$ is the evidence lower bound (ELBO). As the equation indicates, maximizing the ELBO is equivalent to minimizing $\mathcal{D}[Q(z|s_n)||P(z|s_n)]$. To maximize the ELBO, for simplicity, $Q(z|s_n)$ is assumed to be a multivariate Gaussian distribution parameterized by a neural network $h$: $Q(z|s_n) = \mathcal{N}(z|h_\mu(s_n), h_\Sigma(s_n))$, $P(s_{nmk}|z)$ a Bernoulli distribution parameterized by another neural network $g$: $P(s_{nkm}|z) = Bernoulli(s_{nmk}|g(z))$, the prior $P(z)$ the standard multivariate Gaussian distribution.

However, because the term $E_{z\sim Q(z|s_n)}[\log P(s_n|z)]$ in the ELBO involves sampling from the unknown variational distribution $Q(z|s_n)$, we cannot back-propagate the error through the term and therefore there will be no gradient at the term. A popular approach to overcome the issue is to use the 're-parametrization' technique that samples values from the standard multivariate Gaussian distribution $\mathcal{N}(0, I)$ then transforms them into vectors which are similar to the ones sampled from $Q(z|s_n)$. By using the technique, $E_{z\sim Q(z|s_n)}[\log P(s_n|z)]$ can be rewritten as:

$$E_{z\sim Q(z|s_n)}[\log P(s_n|z)] = E_{\epsilon\sim N(0,I)}[\log P(s_n|z = h_\mu(s_n) + h_\Sigma^{1/2}(s_n) * \epsilon)] \quad (7)$$

where $\epsilon$ is the random variable sampled from $\mathcal{N}(0, I)$. Now, the expectation and the parameters of $Q(z|s_n)$ are explicitly linked that allows for computing the respective gradients.

The term $\mathcal{D}[Q(z|s_n)||P(z)]$ in the ELBO can be expanded into the following analytic form:

$$\mathcal{D}[Q(z|s_n)||P(z)] = \frac{1}{2}\left(tr(h_\Sigma(s_n)) + (h_\mu(s_n))^\top(h_\mu(s_n)) - k - \log det(h_\Sigma(s_n))\right) \tag{8}$$

where $k$ is the dimensionality of the distribution. In the training process, to compute the gradients of the ELBO, we first draw a value for $z$ using the re-parameterization technique for each training sample, then compute the gradients of $\log P(s_n|z) - \mathcal{D}[Q(z|s_n)||P(z)]$ at the value. The average of the gradients at many samples converges to the true gradient of the ELBO. The neural networks that decide the parameters of the two distributions, $h$ and $g$, are called 'encoder' and 'decoder', respectively, for the convenience of expression.

## 4. Network structure

In this section we introduce the structures of the encoder and decoder for the proposed model.

### *4.1. Encoder*

The encoder is designed to learn the distributions of the latent representations of users. The input of the encoder for each user is the matrix that holds her aspect-level sentiments on all items $s_n$. We populate the cells of the matrix corresponding to the sentiments on unseen items with zeros. The sentiment matrix is passed to a convolutional layer with a set of filters of size $1 \times K$ (the number of aspects). The filters are designed to extract signals that reflect user aspect-level preferences. The results of the convolutional operations are passed through a max-pooling layer, in order to keep only the most salient signals. The results from the max-pooling are fed into a sequence of two dense layers. The output of the last dense layer is split into two parts, as the

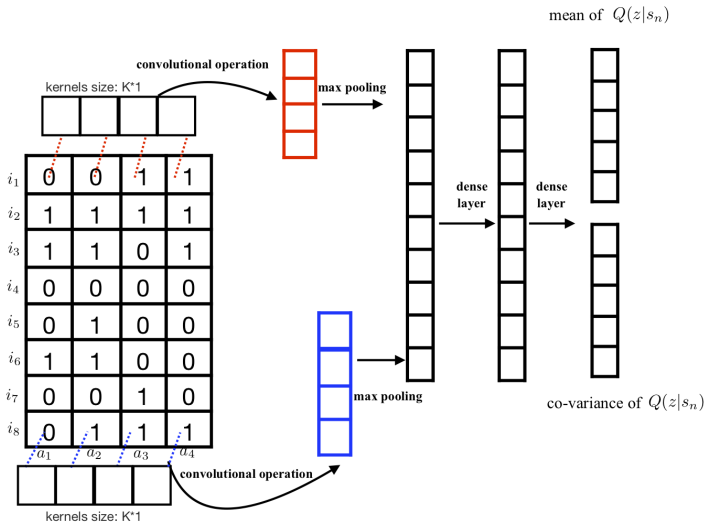
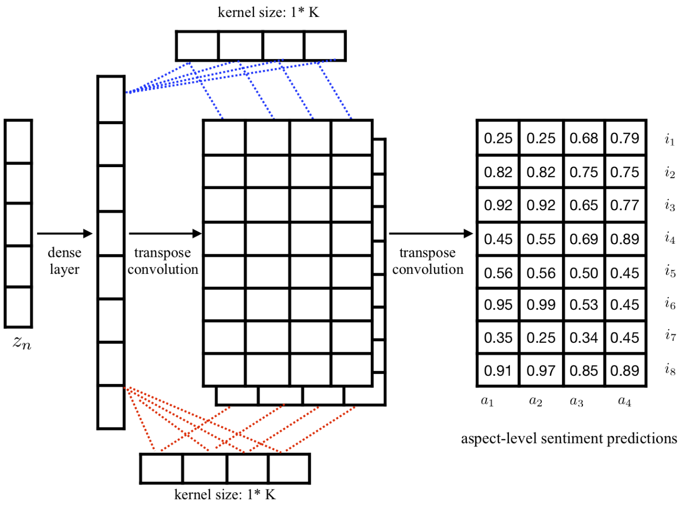


Figure 2: The architecture of the encoder network of VAEARM. The input of the encoder is the binary user aspect-level sentiment matrix, the output is the parameters of the user representation distribution.

mean and covariance matrix of $Q(z|s_n)$, respectively. The architecture of the encoder is shown in Figure 2.

### *4.2. Decoder*

The decoder network is designed to reconstruct user aspect-level sentiment matrices. The input of the decoder is the latent vectors $z_n$ sampled from the re-parametrization process. The latent vector is first fed into a dense layer with hidden units of the number of the items. The weights of the dense layer can be viewed as the encoding of item profiles, and the output of the dense layer as the intermediate results of the interactions between users and items. The dense layer output is then fed into a sequence of two transpose-convolution layers. The first transpose convolutional layer with a set of filters of size $1 * K$, are designed to model the effects of user-item interactions on aspect-level sentiments. The last transpose-convolution layer with a single $1 * K$ filter is designed to combine the results of the previous transpose-convolution layer into a $M * K$ matrix. The logistic function is then applied to each value of the matrix to compute the parameter of the respective Bernoulli distribution over the aspect-level



| $a_1$ | $a_2$ | $a_3$ | $a_4$ | |
|---|---|---|---|---|
| 0.25 | 0.25 | 0.68 | 0.79 | $i_1$ |
| 0.82 | 0.82 | 0.75 | 0.75 | $i_2$ |
| 0.92 | 0.92 | 0.65 | 0.77 | $i_3$ |
| 0.45 | 0.55 | 0.69 | 0.89 | $i_4$ |
| 0.56 | 0.56 | 0.50 | 0.45 | $i_5$ |
| 0.95 | 0.99 | 0.53 | 0.45 | $i_6$ |
| 0.35 | 0.25 | 0.34 | 0.45 | $i_7$ |
| 0.91 | 0.97 | 0.85 | 0.89 | $i_8$ |

Figure 3: The architecture of the decoder network of VAEARM. The input is a user representation vector sampled from $Q(z|s_n)$, the output is the parameters of the Bernoulli distributions over user aspect-level sentiments. Each value in the output indicates the probability of the respective aspect-level sentiment being positive.

sentiments.

### *4.3. Incorporation of overall sentiments*

At the meantime, we incorporate overall sentiments into the proposed model. Naturally, the predicated parameters of the overall sentiment distribution must be highly correlated to that of the aspect-level sentiment distributions. In the SULM model, the authors parameterize the overall sentiment distribution by a linear combination of the parameters of the aspect-level sentiment distributions. In this paper, instead of modeling the relations explicitly, we exploit the strength of the transpose convolution operation in encoding sequential dependency. We simply treat the overall sentiment as a special aspect-level sentiment and arrange it in the last column of the input and output matrices. By this arrangement, the parameters of the overall sentiment distribution become highly dependent upon that of the aspect-level sentiment distributions arranged in the earlier columns of the matrices.

### *4.4. Training and testing*

The neural networks are trained by maximizing the ELBO shown in Equation (3). The input data is aspect-level sentiments of each user $u_n$ on all items. The $m^{th}$ row of the matrix corresponds to the user sentiments on the $K$ aspects of item $i_m$ (it is noteworthy that, as previously mentioned, we treat the overall sentiment as a special aspect-level sentiment and therefore the $K^{th}$ aspect-level sentiment is the overall sentiment). For the items the user has not evaluated, the respective rows of the input matrix are populated with 0s. However, the padding for unseen items may significantly affect the loss of the model and therefore cause unexpected results. To rule out the impacts of the padding, we mask out the respective predictions for the unseen items when computing the loss.

At testing time, for a test user $u_n$, we first draw $V$ samples from the learned distribution $Q(z|s_n)$, then feed each of the samples to the decoder. The decoder will output a $M * K$ matrix for each of the samples. We average the $V$ matrices into a single matrix $y_n$ of shape $M * K$, as the final prediction. In $y_n$, the $(m, k)$ cell holds the parameter of the Bernoulli distribution over the sentiments of $u_n$ on the $k^{th}$ aspect of item $i_m$. In other words $y_{nmk} \in (0, 1)$ is the predicted probability of $s_{nmk}$ being positive. The last column of $y_n$ holds the predicted probabilities for the overall sentiments.

### *4.5. Two-stage item ranking*

The majority of existing aspect-based recommendation models use the predictions for overall sentiments to rank items and select the top items as the recommendations. This approach is not well aligned with the realistic behavior of consumers to make decisions. In a real world situation, to select an item from a large pool of candidates, a user usually first selects the top $X$ items by their overall goodness, and then has close examination on the attributes of the top items to make decisions. We model this behavior with a two-stage ranking scheme. We first rank all items by the overall predictions, namely the values in last column of $y_n$, to select the top 100 items, and then rerank the top items by the predictions for the 'decision-making-related' aspects (Mauro et al., 2021). In this paper we use the predictions for the aspect that is most frequently discussed by all users across a dataset, and the aspect that is most frequently

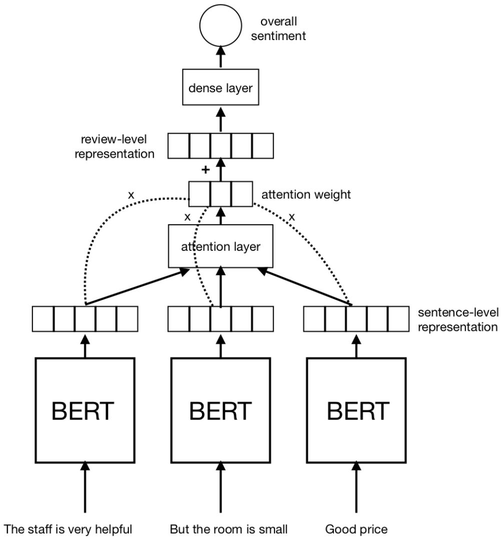


Figure 4: The structure of the SA-BERT model.

discussed by a target user, to rerank the items. The first aspect reflects the common concern of all consumers, the second the personal preference of a target user. We combine the predictions for these aspect-level sentiments and the overall sentiment for the reranking stage as follows:

$$\bar{y}_{nm} = \alpha * y_{nma_c} + \beta * y_{nma_p} + (1 - \alpha - \beta) y_{nma_o} \tag{9}$$

where $a_c \in [1, K]$ denotes the most frequently discussed aspect across the dataset, $a_p \in [1, K]$ the most frequent discussed aspect by a user $u_n$, $a_o$ the overall sentiment. $\alpha, \beta \in [0, 1]$ are hyper-parameters that control the impacts of the predictions for the two aspects on the reranking and can be tuned by cross-validation.

## 5. Sentence-level sentiment classification

In many review datasets, each sample may provide an overall rating on 1-5 scale and a piece of review text, but usually does not provide aspect-level ratings or sentiments. To obtain the aspect-level sentiments, we have to perform sentiment and aspect classification on the review text. In this paper, to extract the aspect-level sentiment information from a review, we analyze each sentence to detect what aspects are discussed in it, and what the sentiments are associated with the aspects. If the number of positive sentences discussing an aspect is larger than that of the negative sentences, then the sentiment for the aspect is assumed to be positive, otherwise negative.

To detect the aspects, for simplicity, we use a keyword based approach that will be introduced in the experiment section. To detect the sentiments of review sentences, we use a supervised sentiment classifier, that takes labeled datasets to train. We can either manually label a large number of review sentences as the training data, or generate the data by simply treating each review as a single long sentence and the overall rating as the sentiment label. We prefer the second way in order to save the labeling costs. However, simply feeding reviews into an existing classifier may suffer low performance because sentence-level sentiment signals are lost in the training process.

In this research we wish to keep the sentence-level signal in the training process, but without sentence-level labeling. To achieve that, we adopt a sentence-level attention mechanism, which is similar as the approach used in the PARADE model (Li et al., 2020). The PARADE model was originally created to estimate the relevance between queries and documents for information retrieval tasks. In the model, to estimate the relevance between a query and a document, the authors first segmented a document into passages, and then passed each passage into the BERT model (Devlin et al., 2019) to obtain passage-level relevance signals. The passage-level signals are aggregated by an attention mechanism into the document-level relevance score. This idea can be easily adapted for training sentence-level classifiers by treating reviews as the 'documents', review sentences as the 'passages' and overall ratings as the 'document-level' labels.

In this research we first segment each review into $T$ sentences, then pass each sentence into the BERT model. For each sentence, the contextual vector associated the respective [CLS] token in the BERT model is used to represent the sentence. The resulting $T$ contextual vectors are fed into a dense layer to compute their attention scores. The contextual vectors are then combined into a single vector by the sum of the vectors weighted by the attention scores. The resulting vector is passed into a dense layer to predict the overall sentiment. The model is trained by minimizing the errors between the predictions and the ground-truth overall sentiments. At testing time, given a test sentence, we compute its [CLS] contextual vector and feed it to the last dense layer (the attention layer is skipped because we are dealing with only one sentence). The output of the dense layer is used as the sentiment prediction for the input sentence. For the convenience of expression, the sentence-level attention approach is called SA-BERT hereinafter. The structure of SA-BERT is shown in Figure 4.

## 6. Experimental setup

### *6.1. Datasets and evaluation metrics*

Table 1: Statistics of the TripAdvisor and Amazon review datasets

| | Hotel | Cellphone | Clothing |
|---|---|---|---|
| Number of authors | 162043 | 27879 | 39387 |
| Number of items | 6419 | 10429 | 23033 |
| Number of reviews | 409239 | 194439 | 278677 |
| Number of reviews per user | 2.53 | 6.97 | 7.07 |
| Number of review per item | 63.75 | 18.64 | 12.10 |
| Average rating | 3.65 | 4.12 | 4.25 |
| Average review length | 180 | 93 | 61 |

In the experiment we use two datasets: TripAdvisor hotel review dataset (Wang et al., 2010) and Amazon product review dataset (McAuley & Leskovec, 2013). In the TripAdvisor dataset, users not only provide overall ratings, but also aspect-level ratings on a 1-5 scale for 9 possible aspects: Business service; Internet access; Check in / front desk; Cleanliness; Location; Rooms; Service; Sleep Quality; Value. The aspect-level ratings, that can serve as a natural and accurate indicator of aspect-level sentiments, are very desirable for our evaluation. It not only saves us the work to perform aspect-level sentiment classification on the review text, but also rules out the impact of inaccurate results from the classification process when comparing the proposed model and baselines. We convert the ratings greater than 3 into the positive sentiment, and otherwise the non-positive sentiment. Also, we are interested in examining the behavior of the proposed model with different training sizes, we randomly draw 50% of the reviews to create a subset of the data. We evaluate the model on the full dataset and the subset separately.

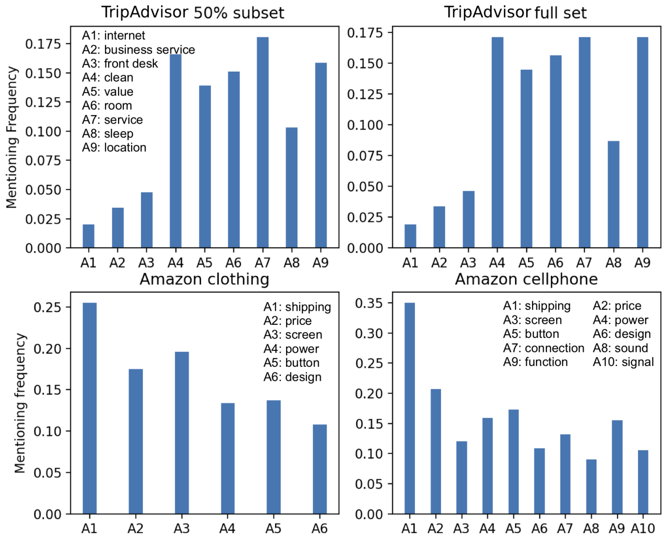


Figure 5: Aspect mentioning frequencies of the TripAdvisor and Amazon datasets

The Amazon dataset contains ratings and reviews for two categories of products: cellphone, clothing. In the dataset, each sample consists of an overall rating on a 1-5 scale and a piece of review text. The overall ratings are converted into the binary sentiments by the same way mentioned above. Aspect-level sentiments or ratings are not readily provided. To obtain that information, we detect aspects and their sentiments from the review text. We predefined the following 10 aspect labels for the cellphone category: shipping; price; screen; power; button; design; connection; sound; function; signal. We predefined the following 6 aspect-level labels for the clothing category: size; price; shipping; material; quality; design. To detect the aspects, we extract the top 100 most frequent nouns from the review text for each product category, and manually associate each of them with an aspect label. A sentence is assigned an aspect label if at least one of the respective nouns occur in the sentence. This approach has high precision rates, but may suffer low recalls: the reviews do not contain any of the top nouns will not be assigned any aspect label. We remove the reviews that are assigned no aspect label from the dataset. To detect the sentiments, we train sentence-level sentiment classifiers for each product category using the SA-BERT model introduced previously. We first train the classifier with reviews on the training split of each product category, then use it to classify the sentiments of review sentences. To evaluate the performance of the approach, we also manually label a set of 2000 sentences for each product category.

Some important statistics of the Amazon and TripAdvisor datasets are shown in Table 1. Also mentioning frequencies of all aspects in the datasets are shown in Figure 5.

### *6.2. Baselines*

For the sentence-level sentiment classification task, the proposed SA-BERT is compared against the following neural classifiers (Minaee et al., 2021) that are also trained under the supervision of overall sentiment labels: BERT; CNN; LSTM. We use the implementations of these models provided by the Tensorflow official website [1].

The aspect-based recommendation tasks consist of two sub-tasks: item ranking; aspect-level sentiment prediction. For item ranking, as previously introduced, we use the two-stage ranking scheme. In the first stage, we rank all items by overall sentiment predictions. In the second stage we rerank the top 100 items from the first stage by combining overall and aspect-level sentiment predictions. We evaluate the proposed model and baselines in each stage. In the first stage, VAEARM is compared against the following models.

- MF: a linear low-rank factorization model that factorizes overall ratings into the products of user and item latent factor matrices.
- VAECF (Liang et al., 2018): a Bayesian neural based model that first projects user sentiments on seen items into a latent space then ‘reconstructs’ the sentiments on unseen items from the latent space.
- LRPPM (Chen et al., 2016): a linear tensor factorization based model that factorizes aspect-level sentiments into the pair-wise products of user- and item- and aspect-level latent factor matrices and learn the matrices by optimizing a ranking loss of user preferences over various aspects.

[1] https://www.tensorflow.org/

- SULM (Bauman et al., 2017): a linear model that employs distinct aspect-level latent factors for each user and item, and assumes the aspect-level sentiments of a user on an item are the results of the interactions between the aspect-level latent factors associated with the user and the item.

- CoSTCo (Liu et al., 2019): a neural based model that uses convolutional neural networks to learn latent factors for users, items and aspects. To use the model, we break the sentiments of a user on the $K$ aspects of an item into $K$ quartets $\{$ user identity, item identity, aspect identity, aspect-level sentiment$\}$. The overall sentiment is treated as an aspect-level sentiment.

- Autoencoder (AE): we concatenate the same encoder and decoder for the proposed VAEARM model to build an autoencoder model.

Among the above baselines, MF and VAECF are general recommendation models. MF is trained on user-item rating matrices, VAECF user-item sentiment matrices. Others are aspect-based recommendation models and trained on user-item-aspect sentiment matrices. In the reranking and aspect-level sentiment prediction, the proposed model is compared only with the above aspect-based recommendation models.

### *6.3. Metrics*

We use the F1 score to measure the performance of the sentence-level sentiment classifiers. For the recommendation models, the performance of each model is evaluated by 5-fold cross validation. Each dataset is split into five folds. Four folds are used as the training set (one random fold among the four folds is used as the validation set for hyper-parameter tuning), and the remaining fold as the test set at a time. For item ranking, the following ranking based metrics (Ricci et al., 2015) are used to measure the performance: normalized discounted cumulative gain (NDCG@K), and mean average precision (MAP). There exist a number of methodologies (Bellogin et al., 2011) to evaluate the ranking performance. This paper adopts the ‘TestItems’ methodology in which each model ranks all items for each user in the test set except the ones she has evaluated in the training set. For the aspect-level sentiment prediction task, the performance is evaluated in terms of ROC AUC score.

### *6.4. Hyper-parameters*

For the BERT classier and SA-BERT, we use BERT-small provided by Tensorflow Hub [2]. The maximum number of review sentences for SA-BERT is set to 10.

The hyper-parameters of all models are fine-tuned on the validation set using NDCG@100 as the objective metric. For MF, the dimensionality of the latent factors is set to [1,50] with a step size of 5, the regularization coefficients for the latent factors [0.01,0.2] with a step size of 0.02. For other baseline models we follow the same hyper-parameter tuning procedures described in the original papers. For the Autoencoder baseline and the proposed VAEARM model, the dimensionality of the latent vector $z$ is set to [500,1000] with a step size of 100, the filter number for the hidden convolution and transpose convolution layers is set to [100,500] with a step size of 100. The RELU function is used as the activation function for all the hidden layers. The dropout rate for all the dense layers is set to [0.05,0.2] with a step size of 0.05. The coefficient $\alpha$ and $\beta$ in the reranking scheme is set to $[0, 1]$ with a step size of 0.2.

All the training and test processes are run on a server with Xeon Processors running at 2.30 GHz, 12.6 Gigabytes of RAM, and Tesla K80 GPU with 12 Gigabytes of memory.

## 7. Results

In this section we first present the sentence-level sentiment classification results, then the results of the item ranking and aspect-level sentiment prediction.

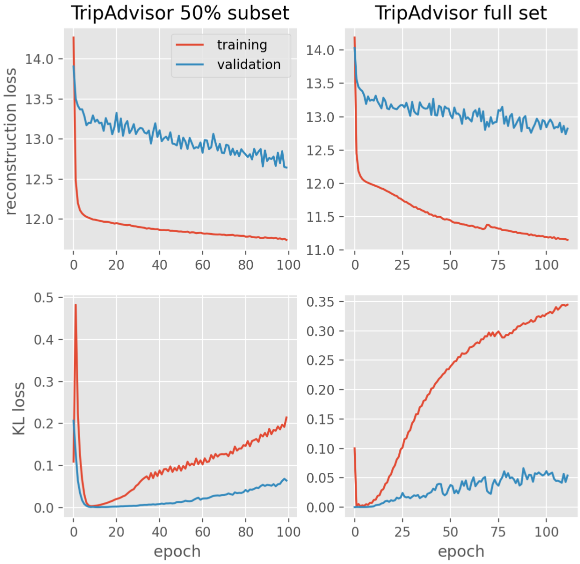
### *7.1. Sentiment classification*

The performance of all the sentiment classifiers trained under the supervision of review-level sentiment labels are shown in Table 2. As shown in the table, first of all, the BERT classifier outperforms both CNN and RNN classifiers by obvious margins. The advantage margin of the BERT classifier on the cellphone reviews is smaller than that on clothing reviews. One reason is that the average length of the cellphone reviews

[2] https://tfhub.dev/google/collections/bert/1

Table 2: Performance of CNN, RNN, BERT and SA-BERT in terms of F1 score on review sentence sentiment classification. The best result in each column is indicated in bold.

|  | Clothing | Cellphone |
|---|---|---|
| CNN | 0.795 | 0.776 |
| RNN | 0.816 | 0.782 |
| BERT | 0.865 | 0.805 |
| SA-BERT | **0.897** | **0.862** |


TripAdvisor 50% subset

TripAdvisor full set

Figure 6: The training curves of VERARM on the TripAdvisor dataset

is much longer than that of the clothing reviews. At least one quarter of the cellphone reviews are longer than 512 tokens and exceed the maximum length BERT can process, therefore they are truncated to fit into the BERT model. The SA-BERT model can solve the problem because it takes each review sentence as the input. SA-BERT enjoys a significant advantage margin of 3.7% and 7.1% over the BERT model on the clothing and cellphone categories, respectively.

### *7.2. Training curves of VAEARM*

The training curves of VAEARM on the TripAdvisor dataset are shown in Figure 6. As shown in the figure, the reconstruction loss keeps decreasing throughout the whole training process. The KL divergence loss quickly decreases to almost 0 at the beginning

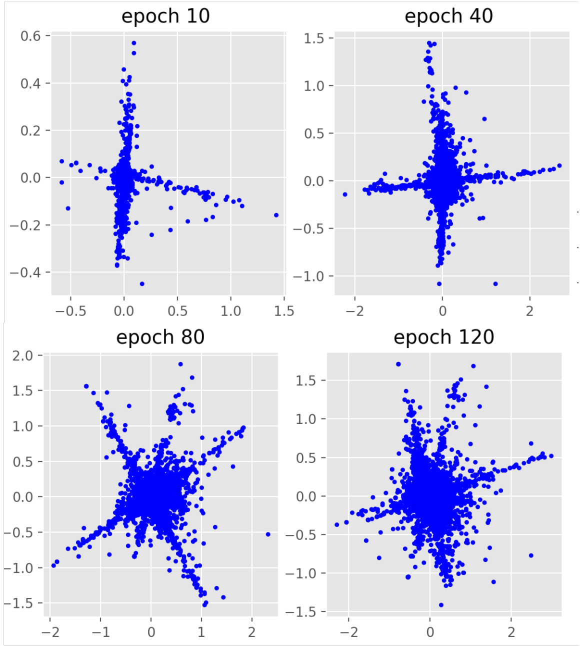


Figure 7: Visualization of the user representations learned by VAEARM at different training stages on the TripAdvisor full set.

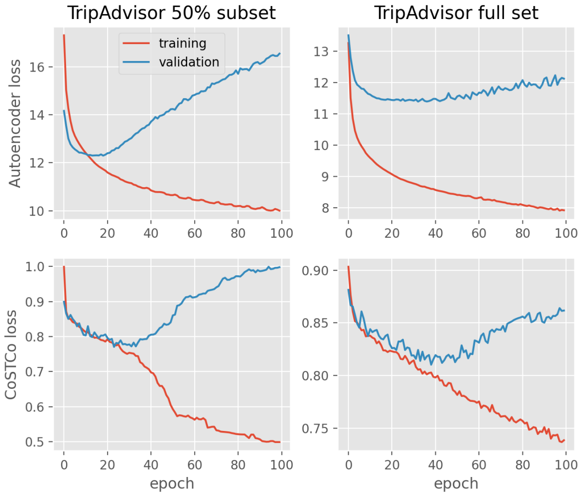


Figure 8: The training curves of Autoencoder and CoSTCo on the TripAdvisor dataset

of the training process, then starts to increase over epoch slowly. This suggests the KL loss is working as a regularizer that imposes penalty on the model as it fits the training data better and better.

On the TripAdvisor full dataset, we use PCA to visualize the means of the variational posterior distributions as the latent representations of users at different stages of the training process in Figure 7. As shown in the figure, at Epoch 10, the data points are askew distributed: the projections of almost all data points at the horizontal axis are 0, indicating the model is underfitting the data and the representations carry very limited information. As the epoch increases, the data points start to spread out and become increasingly normally distributed. However, the shift in the distribution of the data points is constrained. At Epoch 100 the majority of the data points are still distributed around a mean that is not far from the (0,0) point. This shows the effectiveness of the KL divergence loss as a regularizer to prevent overfitting.

We also show the training curves of Autoencoder and CoSTCo in Figure 8 on the TripAdvisor dataset. As shown in the figure, the two models overfit the training data very quickly, especially on the 50% subset.

### 7.3. *Ranking by overall sentiment predictions*

Table 3: Initial ranking results on the TripAdvisor dataset. The best result in each column is indicated in bold. The superscript † and ‡ denote statistical significance (p-value $< 0.05$) over CoSTCo and Autoencoder, respectively.

| Model | 50% subset | | Full set | |
|---|---|---|---|---|
| | NDCG@100 | MAP | NDCG@100 | MAP |
| MF | 0.0941 | 0.0532 | 0.0997 | 0.0566 |
| SULM | 0.1335 | 0.0912 | 0.1612 | 0.1225 |
| LRPPM | 0.1445 | 0.1027 | 0.1582 | 0.1195 |
| VAECF | 0.2151 †‡ | 0.1872†‡ | 0.2312†‡ | 0.1971†‡ |
| CoSTCo | 0.1776 | 0.1421 | 0.1865 | 0.1516 |
| AE | 0.1895† | 0.1502† | 0.2179† | 0.1852† |
| VAEARM | **0.2419**†‡ | **0.2033**†‡ | **0.2614**†‡ | **0.2152**†‡ |

The comparison results on the TripAdvisor dataset are reported in Table 3. First of all, the neural based models outperform the linear models on both the 50% subset and the full set by very clear margins. This can be attributed to the high expressive power of the neural models. Among all the linear models, MF is the least performing model. Among all the neural based models, CoSTCo is the least performing model. Autoencoder outperforms CoSTCo on both the 50% subset and the full set, suggesting the effectiveness of the proposed model architecture in encoding the user-item interactions. On the 50% subset, VAECF and VAEARM outperform Autoencoder by significant margins of 13.4% and 27.6%, respectively; on the full set 6.1% and 19.0%, respectively. This is attributed to the Bayesian approach of these models that prevents them from overfitting. VAEARM outperforms VAECF by a margin of at least 10%, partly thanks to its incorporation of the aspect-level sentiment signals.

Table 4: Initial ranking results on the Amazon datasets. The best result in each column is indicated in bold. The superscript † and ‡ denote statistical significance (p-value $< 0.05$) over CoSTCo and Autoencoder, respectively.

| Model | Cellphone | | Clothing | |
|---|---|---|---|---|
| | NDCG@100 | MAP | NDCG@100 | MAP |
| MF | 0.0855 | 0.0609 | 0.0805 | 0.0599 |
| SULM | 0.0932 | 0.0625 | 0.0852 | 0.0610 |
| LRPPM | 0.0982 | 0.0751 | 0.0926 | 0.0677 |
| VAECF | 0.1625 †‡ | 0.1457†‡ | 0.1545†‡ | 0.1321 †‡ |
| CoSTCo | 0.1429 | 0.1196 | 0.1125 | 0.1062 |
| AE | 0.1577† | 0.1366† | 0.1294† | 0.1152† |
| VAEARM | **0.1894**†‡ | **0.1602**†‡ | **0.1779**†‡ | **0.1572**†‡ |

The results on the Amazon dataset are shown in Table 4. Unlike the results on the TripAdvisor dataset, the performance of SULM and LRPPM is only slightly higher than that of MF. This possibly suggests that SULM and LRPPM are very susceptible to the noise introduced by the aspect and sentiment classification process. The performance of the neural based models is higher than that of the linear models. It is noteworthy that the performance of CoSTCo and Autoencoder on the clothing category is obviously lower than that on the cellphone category. This is possibly caused by that the clothing reviews are much more sparse than the cellphone reviews and therefore the two models suffer worse overfitting on the clothing category. CoSTCo and Autoencoder underperform VAECF and VAEARM. VAEARM still outperforms VAECF by obvious advantage margins on both product categories, despite the noise from the classification process.

### *7.4. Top 100 items reranking*

Table 5: Top 100 items reranking results on the TripAdvisor dataset. The percentage next to the NDCG and MAP values indicates how much improvement the reranking causes from the initial ranking. The superscript † denotes statistical significance (p-value $< 0.05$) over the initial ranking.

| Model | 50% subset | | | | Full set | | | |
|---|---|---|---|---|---|---|---|---|
| | NDCG@100 | | MAP | | NDCG@100 | | MAP | |
| SULM | 0.1379 | 3.3%† | 0.0951 | 4.3%† | 0.1681 | 4.3%† | 0.1268 | 3.5%† |
| LRPPM | 0.1503 | 4.0%† | 0.1076 | 4.8%† | 0.1617 | 2.2%† | 0.1230 | 2.9%† |
| CoSTCo | 0.1897 | 6.8%† | 0.1525 | 7.3%† | 0.2024 | 8.5%† | 0.1625 | 7.2%† |
| AE | 0.2141 | 11.3%† | 0.1694 | 12.8%† | 0.2521 | 15.7%† | 0.2050 | 10.7%† |
| VAEARM | **0.2773** | 14.6%† | **0.2194** | 7.9%† | **0.2994** | 14.5%† | **0.2405** | 11.8%† |

The reranking results of all models are shown in Table 5 and 6. First of all, we observed improvements of the reranking over the initial ranking on both the linear and neural based models. However, the performance improvements on the linear models are less obvious than that on the neural based models. On the TripAdvisor dataset we observed improvements of 2.2%-4.8% on the linear models, and 6.8%-15.7% on the neural based models. On the Amazon dataset, we observed improvements of 1.1%-3.6% on the linear models. However, some of improvements are not statistically significant. By contrast, the reranking of the neural based models on the Amazon dataset improves the initial ranking by obvious margins of 4.1%-12.6%. One possible reason is that, the aspect-level predictions of the linear models are usually similar as the overall predictions, and therefore incorporating the aspect-level predictions of the linear models do not add much information to the ranking by their overall predictions.

Among the neural models, CoSTCo has lower performance gain than Autoencoder and VAEARM. The reranking of Autoencoder and VAEARM enjoys an average improvement of more than 10% from their initial ranking, suggesting the aspect-level predictions of the two encoder-decoded based models are highly complementary to their overall predictions.

Table 6: Top 100 reranking results on the Amazon dataset. The percentage next to NDCG and MAP indicates how much improvement the reranking causes from the initial ranking. The superscript † denotes statistical significance (p-value $< 0.05$) over the initial ranking.

| Model | Cellphone | | | | Clothing | | | |
|---|---|---|---|---|---|---|---|---|
| | NDCG@100 | | MAP | | NDCG@100 | | MAP | |
| SULM | 0.0966 | 3.6%† | 0.0640 | 2.4%† | 0.0859 | 1.1% | 0.0621 | 1.8% |
| LRPPM | 0.0999 | 1.7% | 0.0769 | 2.4% | 0.0949 | 2.5%† | 0.0692 | 2.2% |
| CoSTCo | 0.1492 | 4.4%† | 0.1255 | 4.9%† | 0.1182 | 5.1%† | 0.1106 | 4.1%† |
| AE | 0.1775 | 12.6%† | 0.1490 | 9.1%† | 0.1409 | 8.9%† | 0.1242 | 7.8%† |
| VAEARM | **0.2105** | 11.1%† | **0.1770** | 10.5%† | **0.1920** | 7.9%† | **0.1728** | 9.9%† |

### *7.5. Aspect-level sentiment prediction*

Table 7: Aspect-level sentiment prediction results in terms of AUC score on the TripAdvisor 50% subset. The best result in each row is indicated in bold. In the 'Average' row, the superscript † and ‡ denote statistical significance (p-value $< 0.05$) over CoSTCo and Autoencoder, respectively.

| | VAEARM | LRPPM | SULM | CoSTCo | AE |
|---|---|---|---|---|---|
| Cleanliness | **0.841** | 0.735 | 0.723 | 0.776 | 0.804 |
| Location | **0.772** | 0.717 | 0.631 | 0.736 | 0.732 |
| Rooms | **0.854** | 0.792 | 0.781 | 0.801 | 0.832 |
| Service | **0.832** | 0.747 | 0.717 | 0.775 | 0.809 |
| Value | **0.736** | 0.679 | 0.662 | 0.702 | 0.715 |
| Front desk | **0.747** | 0.726 | 0.728 | 0.735 | 0.733 |
| Sleep | **0.675** | 0.625 | 0.634 | 0.659 | 0.667 |
| Business service | 0.616 | **0.621** | 0.595 | 0.576 | 0.601 |
| Internet | 0.620 | **0.626** | 0.617 | 0.557 | 0.605 |
| Average | **0.744**†‡ | 0.696 | 0.676 | 0.702 | 0.722 † |

The comparison results on the TripAdvisor dataset are reported in Table 7, 8. In general, the average AUC scores of the neural models are higher than that of the linear models. However, on the internet aspect and the business service aspect, which are rarely discussed in the dataset, the performance of the neural based CoSTCo and Autoencoder is obviously lower than that of the linear LRPPM model. This indicates their high tendency to overfitting on sparse data. The Autoencoder model outperforms CoSTCo, further showing the effectiveness of the proposed model architecture in encoding aspect-level sentiment signals. The proposed VAEARM model has the highest average AUC scores among all the models. On the two rare aspects, the proposed

Table 8: Aspect-level sentiment prediction results in terms of AUC score on the TripAdvisor full dataset. The best result in each row is indicated in bold. In the 'Average' row, the superscript † and ‡ denote statistical significance (p-value $< 0.05$) over CoSTCo and Autoencoder, respectively.

| | VAEARM | LRPPM | SULM | CoSTCo | AE |
|---|---|---|---|---|---|
| Cleanliness | **0.861** | 0.747 | 0.742 | 0.806 | 0.837 |
| Location | **0.805** | 0.732 | 0.647 | 0.747 | 0.762 |
| Rooms | **0.871** | 0.803 | 0.809 | 0.837 | 0.859 |
| Service | **0.850** | 0.769 | 0.738 | 0.805 | 0.837 |
| Value | **0.765** | 0.692 | 0.672 | 0.722 | 0.733 |
| Front desk | **0.770** | 0.735 | 0.739 | 0.759 | 0.755 |
| Sleep | **0.712** | 0.647 | 0.659 | 0.681 | 0.699 |
| Business service | **0.654** | 0.642 | 0.622 | 0.585 | 0.614 |
| Internet | **0.650** | 0.639 | 0.641 | 0.602 | 0.621 |
| Average | **0.771**$^{\dagger\ddagger}$ | 0.711 | 0.697 | 0.727 | 0.746$^{\dagger}$ |

model has similar performance as the linear models. On the most frequently discussed aspects, such as cleanliness and service, the proposed model enjoys the most obvious advantage margins over the linear baselines.

Table 9: Aspect-level sentiment prediction results in terms of AUC score on the Amazon cellphone category. The best result in each row is indicated in bold. In the 'Average' row, the superscript † and ‡ denote statistical significance (p-value $< 0.05$) over CoSTCo and Autoencoder, respectively.

| | VAEARM | LRPPM | SULM | CoSTCo | AE |
|---|---|---|---|---|---|
| Shipping | **0.772** | 0.636 | 0.624 | 0.702 | 0.735 |
| Price | 0.775 | 0.726 | 0.705 | 0.742 | **0.786** |
| Screen | **0.803** | 0.726 | 0.716 | 0.752 | 0.765 |
| Power | 0.776 | 0.721 | 0.726 | 0.764 | **0.779** |
| Button | **0.789** | 0.728 | 0.715 | 0.766 | 0.780 |
| Design | **0.785** | 0.725 | 0.693 | 0.762 | 0.755 |
| Sound | **0.773** | 0.703 | 0.653 | 0.753 | 0.747 |
| Connection | **0.742** | 0.685 | 0.696 | 0.726 | 0.722 |
| Functionality | **0.756** | 0.698 | 0.647 | 0.746 | 0.739 |
| Signal | **0.782** | 0.712 | 0.706 | 0.745 | 0.759 |
| Average | **0.775**$^{\dagger\ddagger}$ | 0.706 | 0.688 | 0.746 | 0.757$^{\dagger}$ |

The comparison results on the Amazon datasets are reported in Table 9 and Table 10. We observed similar results. The VAEARM model has the highest average AUC scores across both product categories. Similarly as on the TripAdvisor dataset, the

Table 10: Aspect-level sentiment prediction results in terms of AUC score on the Amazon clothing category. The best result in each row is indicated in bold. In the 'Average' row, the superscript † and ‡ denote statistical significance (p-value $< 0.05$) over CoSTCo and Autoencoder, respectively.

| | VAEARM | LRPPM | SULM | CoSTCo | AE |
|---|---|---|---|---|---|
| Size | **0.853** | 0.701 | 0.705 | 0.735 | 0.746 |
| Price | **0.795** | 0.706 | 0.721 | 0.725 | 0.749 |
| Shipping | **0.798** | 0.697 | 0.701 | 0.719 | 0.752 |
| Material | **0.808** | 0.739 | 0.728 | 0.732 | 0.769 |
| Quality | **0.766** | 0.697 | 0.685 | 0.722 | 0.749 |
| Design | 0.732 | 0.676 | 0.652 | 0.710 | **0.743** |
| Average | **0.792**†‡ | 0.702 | 0.697 | 0.724 | 0.751† |

proposed model enjoys the most obvious margins over the linear models on shipping in the cellphone category and size in the clothing category, which are also the most frequently discussed aspects.

## 8. Discussion

In the aspect-level sentiment prediction subtask, the experiment results show that VAEARM enjoys significant performance advantages over the linear SULM and LRPPM models on frequently discussed aspects. We found that in reviews the ground truth sentiments on these aspects are more likely than that on other aspects to be different from the respective overall sentiments. For example, in the Amazon cellphone review dataset, there are many cases where the sentiments on shipping are positive whereas the overall sentiments are negative. There may exist non-linear relations between aspect-level sentiments and overall sentiments. The linear SULM and LRPPM are ineffective to capture the relations. In the experiment we found that these linear models always give consistent predictions for aspect-level and overall sentiments. In other words they are likely to give either all positive or all negative predictions for aspect-level and overall sentiments. Therefore, the performance of the linear models is low on the test samples whose ground truth aspect-level and overall sentiments are inconsistent.

In a typical encoder-decoder based model, a key hyper-parameter is the dimensionality of the latent encoding vector $z$. However, in the experiment we found that the

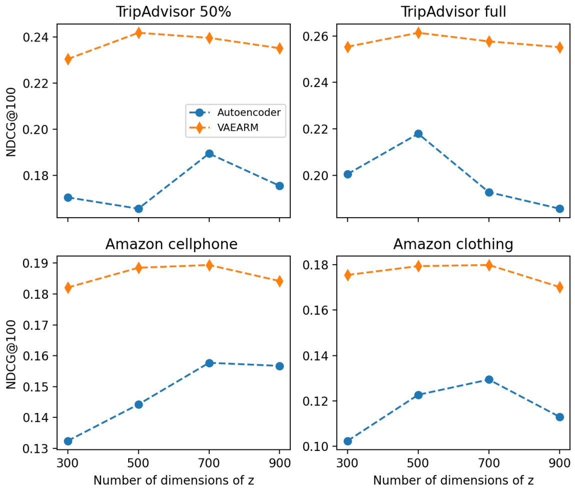


Figure 9: The ranking performance of VAEARM and Autoencocder with different latent vector dimensionalities.

proposed model is much less susceptible than the Autoencoder model to the hyper-parameter. We show the performance of VAEARM and Autoencoder in item ranking with different values for the hyper-parameter in Figure 9. As shown in the figure, the performance of the proposed model is relatively stable across the entire value range, whereas the performance of the Autoecnder varies greatly. This possibly suggests that the Bayesian approach not only reduces the risk of overfitting, but also hyper-parameter sensitivity. In a realistic engineering situation, low sensitivity is always preferred as it can significantly reduce hyper-parameter tuning efforts.

## 9. Conclusion

### *9.1. Theoretical and practical implications*

Overall, this research advances the state of the art in explainable recommendation systems. The proposed model can serve as a strong alternative to existing linear and neural based aspect-based recommendation models, especially when the training data is scarce and sparse. The experiment results show that the proposed model outperforms a number of strong baselines in item ranking and aspect-level sentiment prediction, thanks to its Bayesian approach and high expressive power.

Aspect-based recommendation models give both overall and aspect-level sentiment predictions for unseen items. Existing works usually rank items only by the overall sentiment predictions to generate recommendations. The aspect-level predictions are rarely explored for improving the item ranking performance. This research bridges the gap by introducing a two-stage ranking scheme that first retrieves top items by overall sentiment predictions, then reranks the items by combining the aspect-level and overall sentiment predictions. We observed obvious improvements of the two-stage ranking scheme over the ranking by the overall sentiment predictions, not only on the proposed model, but also on the baselines. This implies the effectiveness and wide applicability of the two-stage ranking scheme.

The incorporation of the Bayesian approach and the two-stage ranking scheme differentiate the proposed model from the existing aspect-based recommendation models. Also, we adopt a sentence-level attention mechanism that allows for using reviews as the training samples, and overall ratings as the sentiment labels to train sentence-level sentiment classifiers. Though the idea is borrowed from the PARADE model, this research demonstrates a practical solution to overcome the low availability of labelled data for training review sentence sentiment classifiers, which play an important role in many research topics related to recommendation systems.

### *9.2. Limitation and future work*

The input of the proposed model is user-item-aspect sentiment matrices and we employ convolution operations to extract user representations from the matrices. As the number of items increases, the size of the matrices will also increase. This possibly leads to exponential increase in computation complexity. To address the issue, we will consider alternative model structures which are much computationally lighter under the framework of the Bayesian approach.

Also, to use the proposed model, it requires aspect-level sentiment classification on review text to build user-item-aspect sentiment matrices as the input. The classification process itself is a non-trivial task and may also introduce noise to the training process. In the future we will consider to incorporate review text into the proposed model and learn the sentiment signals directly from review text, instead of from the matrices which are expensive to obtain.

## References

Bauman, K., Liu, B., & Tuzhilin, A. (2017). Aspect Based Recommendations: Recommending Items with the Most Valuable Aspects Based on User Reviews. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 717–725). Halifax NS Canada: ACM.

Bellogin, A., Castells, P., & Cantador, I. (2011). Precision-oriented evaluation of recommender systems: an algorithmic comparison. In *Proceedings of the fifth ACM conference on Recommender systems - RecSys '11* (p. 333). Chicago, Illinois, USA: ACM Press.

Chambua, J., & Niu, Z. (2020). Review text based rating prediction approaches: preference knowledge learning, representation and utilization. *Artificial Intelligence Review*, (pp. 1–30). Publisher: Springer.

Chen, H., & Li, J. (2020). Neural Tensor Model for Learning Multi-Aspect Factors in Recommender Systems. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence* (pp. 2449–2455). Yokohama, Japan: International Joint Conferences on Artificial Intelligence Organization. doi:10.24963/ijcai.2020/339.

Chen, L., & Wang, F. (2013). Preference-based clustering reviews for augmenting e-commerce recommendation. *Knowledge-Based Systems*, *50*, 44–59.

Chen, X., Qin, Z., Zhang, Y., & Xu, T. (2016). Learning to Rank Features for Recommendation over Multiple Categories. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval* (pp. 305–314). Pisa Italy: ACM.

Chin, J. Y., Zhao, K., Joty, S., & Cong, G. (2018). ANR: Aspect-based Neural Recommender. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management* (pp. 147–156). Torino Italy: ACM.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (pp. 4171–4186). Minneapolis, Minnesota: Association for Computational Linguistics. doi:10.18653/v1/N19-1423.

Diao, Q., Qiu, M., Wu, C.-Y., Smola, A. J., Jiang, J., & Wang, C. (2014). Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS). In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 193–202). New York New York USA: ACM.

Dong, R., Schaal, M., O'Mahony, M. P., McCarthy, K., & Smyth, B. (2013). Opinionated product recommendation. In *International conference on case-based reasoning* (pp. 44–58). Springer.

Frolov, E., & Oseledets, I. (2017). Tensor methods and recommender systems. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *7*, e1201. Publisher: Wiley Online Library.

Hernández-Rubio, M., Cantador, I., & Bellogín, A. (2019). A comparative analysis of recommender systems based on item aspect opinions extracted from user reviews. *User Modeling and User-Adapted Interaction*, *29*, 381–441. Publisher: Springer.

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, *42*, 30–37. Publisher: IEEE.

Li, C., Yates, A., MacAvaney, S., He, B., & Sun, Y. (2020). PARADE: Passage Representation Aggregation for Document Reranking. *arXiv:2008.09093 [cs]*, . URL: http://arxiv.org/abs/2008.09093. ArXiv: 2008.09093.

Liang, D., Krishnan, R. G., Hoffman, M. D., & Jebara, T. (2018). Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 world wide web conference* (pp. 689–698).

Liu, H., He, J., Wang, T., Song, W., & Du, X. (2013). Combining user preferences and user opinions for accurate recommendation. *Electronic Commerce Research and Applications*, *12*, 14–23.

Liu, H., Li, Y., Tsang, M., & Liu, Y. (2019). CoSTCo: A Neural Tensor Completion Model for Sparse Tensors. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 324–334). Anchorage AK USA: ACM. doi:10.1145/3292500.3330881.

Liu, P., Zhang, L., & Gulla, J. A. (2020). Dynamic attention-based explainable recommendation with textual and visual fusion. *Information Processing & Management*, *57*, 102099. doi:10.1016/j.ipm.2019.102099.

Mauro, N., Ardissono, L., & Petrone, G. (2021). User and item-aware estimation of review helpfulness. *Information Processing & Management*, *58*, 102434. doi:10.1016/j.ipm.2020.102434.

McAuley, J., & Leskovec, J. (2013). Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems* (pp. 165–172). Hong Kong China: ACM.

Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., & Gao, J. (2021). Deep Learning–based Text Classification: A Comprehensive Review. *ACM Computing Surveys*, *54*, 1–40. doi:10.1145/3439726.

Musto, C., de Gemmis, M., Semeraro, G., & Lops, P. (2017). A multi-criteria recommender system exploiting aspect-based sentiment analysis of users' reviews. In *Proceedings of the eleventh ACM conference on recommender systems* (pp. 321–325).

Nilashi, M., bin Ibrahim, O., Ithnin, N., & Sarmin, N. H. (2015). A multi-criteria collaborative filtering recommender system for the tourism domain using Expectation Maximization (EM) and PCA–ANFIS. *Electronic Commerce Research and Applications*, *14*, 542–562. Publisher: Elsevier.

Pan, Y., Huo, Y., Tang, J., Zeng, Y., & Chen, B. (2021). Exploiting relational tag expansion for dynamic user profile in a tag-aware ranking recommender system. *Information Sciences*, *545*, 448–464. Publisher: Elsevier.

Ricci, F., Rokach, L., & Shapira, B. (Eds.) (2015). *Recommender Systems Handbook*. Boston, MA: Springer US.

Salakhutdinov, R., & Mnih, A. (2008). Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. In *Proceedings of the 25th international conference on Machine learning* (pp. 880–887).

Wang, H., Lu, Y., & Zhai, C. (2010). Latent aspect rating analysis on review text data: a rating regression approach. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 783–792).

Wu, Y., & Ester, M. (2015). FLAME: A Probabilistic Model Combining Aspect Based Opinion Mining and Collaborative Filtering. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining* (pp. 199–208). Shanghai China: ACM.

Yang, C., Chen, X., Liu, L., & Sweetser, P. (2021). Leveraging semantic features for recommendation: Sentence-level emotion analysis. *Information Processing & Management*, *58*, 102543. doi:10.1016/j.ipm.2021.102543.