Noname manuscript No.
(will be inserted by the editor)

CAESAR: Context-Aware Explanation based on Supervised Attention for Service Recommendations

Lei Li · Li Chen · Ruihai Dong

Received: date / Accepted: date

Abstract Explainable recommendations have drawn more attention from both academia and industry recently, because they can help users better understand recommendations (i.e., why some particular items are recommended), therefore improving the persuasiveness of the recommender system and users' satisfaction. However, little work has been done to provide explanations from the angle of a user's contextual situations (e.g., companion, season, and destination if the recommendation is a hotel). To fill this research gap, we propose a new context-aware recommendation algorithm based on supervised attention mechanism (CAESAR), which particularly matches latent features to explicit contextual features as mined from user-generated reviews for producing context-aware explanations. Experimental results on two large datasets in hotel and restaurant service domains demonstrate that our model improves recommendation performance against the state-of-the-art methods and furthermore is able to return feature-level explanations that can adapt to the target user's current contexts.

Keywords Explainable recommendation · Context-aware recommender systems · Neural network · Multi-task learning

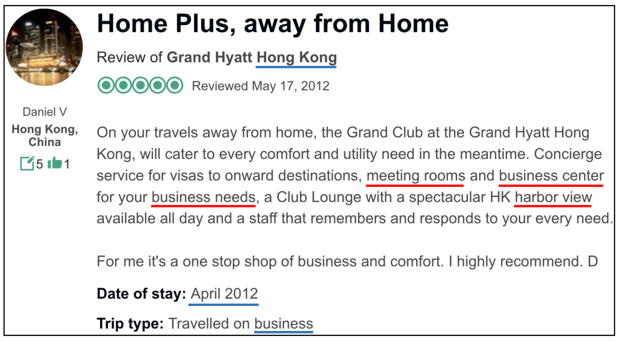
1 Introduction

Recommendation algorithms, such as collaborative filtering [34] and matrix factorization [32], have been widely used in academia and industry to return personalized information or service to users. On one hand, in order to provide more accurate

Lei Li
Department of Computer Science, Hong Kong Baptist University, Hong Kong, China
E-mail: csleili@comp.hkbu.edu.hk

Li Chen
Department of Computer Science, Hong Kong Baptist University, Hong Kong, China
E-mail: lichen@comp.hkbu.edu.hk

Ruihai Dong
Insight Centre for Data Analytics, University College Dublin, Dublin, Ireland
E-mail: ruihai.dong@ucd.ie


Home Plus, away from Home

Review of Grand Hyatt Hong Kong

Reviewed May 17, 2012

Daniel V
Hong Kong, China
5 1

On your travels away from home, the Grand Club at the Grand Hyatt Hong Kong, will cater to every comfort and utility need in the meantime. Concierge service for visas to onward destinations, meeting rooms and business center for your business needs, a Club Lounge with a spectacular HK harbor view available all day and a staff that remembers and responds to your every need.

For me it's a one stop shop of business and comfort. I highly recommend. D

Date of stay: April 2012

Trip type: Travelled on business

Fig. 1 A hotel review example from TripAdvisor, where "contexts" are highlighted with blue lines and "contextual features" (i.e., features relevant to the context) with red lines.

recommendations that adapt to the needs of users in different contextual scenarios, context-aware recommender systems (CARS) have been extensively studied [2,31,25,15,43,7,23]. On the other hand, explainable recommendation, which aims to answer why a particular item is recommended, has drawn increasing attention in recent years [46,14,5,30,4,6,41,39,40,13,9,42,26,24,45]. As a matter of fact, appropriate explanations can help users make better and faster decisions, increase their trust in the system, and/or convince them to try or buy the recommended item [37]. However, few recommendation models have linked the two branches of work for providing *context-aware explanations*. For instance, one popularly used explanation "*You might be interested in [feature], on which this product performs well*" [46,39,13] does not reflect the recommendation's suitability for users' changing needs under different contexts, which however is especially important when users search for a service-oriented product (e.g., hotel and restaurant).

Therefore, to produce informative explanations, as well as accurate recommendations, which fit user context (that refers to "any information that can be used to characterize the situation of an entity" [1], such as companion, season, and destination), we propose a novel neural model, called CAESAR standing for Context-Aware Explanation based on Supervised Attention for Service Recommendations. It particularly harnesses user-generated reviews, because they contain rich contextual features commented by users based on their experiences. It is able to select the user's most concerned context as well as its most relevant features to produce **context-aware feature-level explanation** for the recommendation, i.e., "*This product is recommended to you, because its [features] are suitable for your current [context].*"

To produce such explanation, we need to resolve several challenging issues. The first issue is how to mine high-quality contextual features from user reviews (like the ones highlighted with red lines in Figure 1). For example, pre-defining some contextual seed words and searching for their synonyms [7] may limit the diversity of the extracted features. The second challenging issue is about how to find relevant features for different contexts, because users may care about different features under different situations. For instance, users having a business trip may choose a hotel that provides facilities such as wifi and meeting room (see Figure 1), while couples may prefer romantic dinner. Third, how to find a user's most concerned context is also challenging, since not all of the contexts might be equally important to her/him. For example, still in Figure 1, the user only cared about features related to *trip goal* ("business") and *location* ("Hong Kong"), but not to *time*. As for related work on CARS, though the popularly applied approach based on tensor factorization [19,39] can unify users, items, and contexts into one model, it is incapable of modeling contextual features, since that would increase the dimension of tensor and make it highly sparse. Last but not least, it is essential to guarantee the selected features to match to the user's own preference so that the explanation can be personalized.

In this work, we have proposed CAESAR to address these limitations. To the best of our knowledge, this is the first work that has explicitly considered different types of user contexts (e.g., companion, destination, and time) to generate the feature-level explanation, as well as improving the recommendation accuracy at the same time. Our main contributions are listed below:

- CAESAR can model explicit contextual features via **supervised attention mechanism** to make the selected features match to a user's preference. To that end, we develop a new feature mining approach to discover contextual features from user reviews.
- We further propose a multi-level attention mechanism, i.e., **feature-level** and **context-level**, to adaptively distinguish the importance of different contexts and their related contextual features to both recommendation and explanation.
- Through the experiments on two large real-world datasets (TripAdvisor and Yelp), we show that our model CAESAR consistently outperforms the state-of-the-art methods in terms of recommendation accuracy.
- Moreover, our feature mining approach can find high-quality contextual features from user reviews, which enables our model to produce **context-aware feature-level explanations** that are perceived more helpful to users than context-unaware explanations, as shown in the human evaluation.

The remaining content is organized as follows. Section 2 summarizes related work. We then formulate our problem in Section 3, and in detail present the proposed method in Sections 4 and 5. The experiments and results are discussed in Sections 6 and 7. We make the final conclusions in Section 8.

2 Related Work

There are two lines of research closely related to our work: Context-aware recommender systems (CARS) and explainable recommendation. In this section, we give a brief summary of the two branches of research work.

2.1 Context-Aware Recommender Systems

The existing work on CARS can be classified into two groups. The first group of work has been engaged in leveraging contextual features for recommendation [7,23], for which the recommendation process normally contains several steps, including *contextual feature extraction*, *sentiment orientation detection*, *review score calculation*, and *item score computation*. However, the whole process may involve costly human labeling efforts or require domain knowledge. In comparison, we aim to build a general neural network for modeling contextual features after the feature mining process, so as to reduce human efforts. The other difference is that our model can utilize contextual features for both recommendation and explanation, while those related methods mainly leverage contextual features for recommendation only.

The second group of work does not consider contextual features, but models contexts in various ways. For example, in [19], users, items, and contexts are all represented as factors in a user-item-context tensor, which is then factorized by Tucker decomposition [21]. In [28], tensors are also applied, but used to derive contextual operating matrices and further context-specific vectors for rating prediction. [3] extends matrix factorization [32] by taking the influences of contexts as bias parameters, which could be related to items or their categories. In factorization machines (FM) [33] based methods [43,15,44], users, items, and contexts are equally treated as features for sparse data prediction. Specifically, [43] leverages attention mechanism to distinguish the importance of different feature interactions. [15] applies neural networks to learn non-linear and high-order feature interactions. [44] performs 3D convolutions on an interaction cube to capture high-order interaction signals. More recent CARSs have attempted to characterize the relations between users/items and contexts in an attentive manner to distinguish the impacts of different contexts on the user's preference [31,25]. The major difference between [31] and [25] is that the former adopts vanilla attention mechanism, while the latter utilizes co-attention.

However, although these methods achieve certain accuracy improvement, they are hardly explainable because the features are mostly in latent representations (e.g., vectors or scalars), while our proposed approach can leverage explicit contextual features for achieving both recommendation and explanation.

2.2 Explainable Recommendation

We divide related studies on explainable recommendations into two general groups: Keywords-oriented explanation and natural language explanation.

There are many methods that have produced keyword explanations for recommendations, where keywords can be features extracted from user reviews [46, 14,39], tags of users [41,38], or nodes on knowledge graphs [42,13]. The methods proposed to generate the keyword explanation include matrix factorization [46], tensor factorization [39], graph-based model [14], and neural networks [41,13,42, 36,29]. In particular, the attention mechanism is widely used in neural networks to highlight words in user reviews [36,29], identify some important features from a feature collection [13], or find paths of features on a knowledge graph [42] or on decision trees [41]. Our work also lies in this category because it makes use of attention mechanism, but it differs from related methods in that it produces context-aware feature-level explanations.

Recently, more and more efforts have been put to produce natural language explanations, for which there are two typical ways, i.e., selecting texts written by other users [6,40,9,5], or automatically generating texts [11,30,27,26,24]. For the first one, some work has selected reviews of an item as explanations via either attention mechanism [6] or similarity between two embeddings [5]. As a user review can contain noisy information that is not really useful [17,40], some studies have

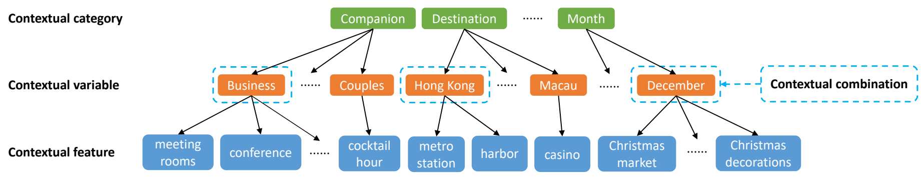


Fig. 2 Context-related concepts and corresponding examples.

attempted to select sentences from reviews of an item [40,9]. For the second, automatic text generation techniques have been leveraged. For instance, long short-term memory (LSTM) [18] and gated recurrent unit (GRU) [10] have been adopted to predict the textual review [11,30] or tip [27] that a user would write for an item. However, as pointed out by [48], these techniques tend to generate safe and universal texts that carry little meaning and thus may be less useful to users.

The works closely related to ours are [9,4,35], because they have also considered contexts. However, [9] and [4] respectively provide time sequence based and location based explanations, so their models are restricted to only one type of context (i.e., time or location), while our model can accommodate various types of context into explanation. The third work [35] is limited to item-level context-aware explanation, e.g., "*recommend for use as [couples]*", so the explanation cannot reason in terms of what features the recommended item would be suitable for the user's current contexts.

3 Problem Formulation

Before stating the research problem, we introduce some important concepts that will be used in the following content. As shown in Figure 2, contexts can be grouped into different categories that we call **contextual categories**, such as *companion*, *destination*, and *month*. We denote them as $\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_m$, where m is the total number of contextual categories. Each contextual category consists of multiple values, e.g., $\underline{\textit{family}}$ and $\underline{\textit{couples}}$ for *companion*. Since a user-item pair normally has one unique value for a contextual category, this **contextual variable** for $\mathcal{C}_j$ (where $j \in \{1, 2, ..., m\}$) is denoted as c_j for brevity. Following [31], we name the combination of contextual variables $(c_1, c_2, ..., c_m)$ for a user-item pair as **contextual combination c**. In this paper, we interchangeably call **c contexts**. As illustrated in Figure 2, there are some features that are highly relevant to a contextual variable, e.g., "*meeting rooms*" and "*conference*" to $\underline{\textit{business}}$. We term such features **contextual features** and denote their collection for a contextual variable c from $\mathcal{C}_j$ as $\mathcal{F}_j^c$.

The goal of our recommendation task is to predict a rating $\hat{r}_{u,i,\mathbf{c}}$, given a user u, an item i, and the corresponding contexts $\mathbf{c}$. Moreover, our proposed model will select contextual features relevant to the user's most concerned context for explanation. At the training stage, the training data consist of users, items, contexts, and contextual features. The key notations and concepts used in our paper are presented in Table 1. We use $\mathbf{p}_u \in \mathbb{R}^d$ to represent the embedding of user $u \in \mathcal{U}$, and $\mathbf{q}_i \in \mathbb{R}^d$ for the embedding of item $i \in \mathcal{I}$. For the j-th context in contextual combination $\mathbf{c}$, i.e., c_j, we denote its embedding as $\mathbf{k}_j \in \mathbb{R}^d$, and the embedding

Table 1 Key notations and concepts.

Symbol	Description
$\mathcal{T}$	training set
$\mathcal{U}$	set of users
$\mathcal{I}$	set of items
$\mathcal{C}_j$	set of values for j-th contextual category
$\mathcal{F}_j^c$	set of features for context c in category j
$\mathbf{p}_u$	embedding of user u
$\mathbf{q}_i$	embedding of item i
$\mathbf{c}$	contextual combination
$\mathbf{k}_j$	embedding of j-th context in $\mathbf{c}$
$\mathbf{H}_j$	embedding matrix of features for j-th context in $\mathbf{c}$
$\mathbf{s}_j$	distribution of features for j-th context in $\mathbf{c}$
$\mathbf{W}$	weight matrix
$\mathbf{w}, \mathbf{b}$	weight vector
w, b	weight scalar
m	number of contextual category
n	number of contextual feature
d	dimension of embedding
$r_{u,i,\mathbf{c}}$	rating assigned by user u on item i under contexts $\mathbf{c}$
$\hat{r}_{u,i,\mathbf{c}}$	predicted rating
$\sigma(\cdot)$	activation function

matrix of contextual features related to this context $\mathbf{H}_j \in \mathbb{R}^{d \times n}$, where d and n respectively represent the dimension of embedding and the number of contextual features.

In Section 4, we first introduce our approach to mine contextual features from user reviews, and then describe our proposed model CAESAR in Section 5 that is developed to achieve both context-aware recommendation and explanation.

4 Contextual Feature Mining

Since some features may not be relevant to certain contextual variables (e.g., feature "*hotel*" may be too general to context *business*), it is necessary to find context-relevant features of each contextual variable for producing better explanation. Our approach for mining those features is comprised of two major steps: extracting features from user reviews, and measuring the relevance between features and contexts. For the former, a sentiment analysis toolkit [47] can be applied to accomplish it. However, the second step is challenging, as it is not intuitive to assign each extracted feature an appropriate weight to indicate its relevance to a particular context. To address this challenge, we have revised the weight assigning strategy originally proposed in [23] to identify high-quality contextual features. Specifically, we utilize point-wise mutual information (PMI), instead of raw occurrence frequency in [23], in order to distinguish the relative importance weights of a feature with respect to different contexts. We then compute the weight of a feature by comparing its relevance degrees to different contexts in the same contextual category, which is also different from [23] as it just computes the weight for all contexts regardless of their categories (for example, they treated *family* and *December* equally without considering their respective categories *companion* and *month*).

To be more concrete, after extracting a list of (feature, opinion, sentiment polarity) entries from textual reviews via the sentiment analysis tool [47] (e.g., (harbor view, spectacular, +1) from the sentence "*a spectacular HK harbor view available all day*"), we first filter out negative features (because their occurrence is much less than that of positive features, which may cause data imbalance issue if we integrate them) and infrequently occurring features. We then aim at discovering the most relevant features to each context associated with the target review by analyzing the relation between them and the context. To this end, we first count a feature f's overall occurrence frequency $freq_f^c$ in the user reviews pertaining to context c in the contextual category $\mathcal{C}_j$, where $j \in \{1, 2, ..., m\}$. To account for the relative importance of a feature to different contexts, we compute the PMI value:

$$PMI_f^c = \frac{freq_f^c}{freq_f \cdot freq^c} \tag{1}$$

where $freq_f$ denotes the total frequency of feature f in all the user reviews and $freq^c$ indicates the total number of features in the user reviews pertinent to context c.

With the PMI values, we calculate the mean value for each feature f over all contexts in $\mathcal{C}_j$ as $avg_f = \sum_{c \in \mathcal{C}_j} PMI_f^c / |\mathcal{C}_j|$, based on which we estimate the statistical error of feature f for context c as $err_f^c = PMI_f^c - avg_f$. Then, we measure the relevance between a feature f and a context c via a weight:

$$w_f^c = \left| err_f^c \right| \tag{2}$$

The larger the weight w_f^c is, the less general the feature f is to the context c, indicating it is more relevant to this context. For each context, we rank all the features according to their weights. Moreover, the ranking positions of features that have the same weight for a certain context are adjusted in accordance with their appearing frequencies under this context.

Lastly, for each context c in the contextual category $\mathcal{C}_j$, we select top n ranked features to construct the contextual feature set $\mathcal{F}_j^c$.

5 Context-Aware Recommendation and Explanation

The results of the previous section are leveraged to generate context-aware recommendation and explanation. In our model CAESAR, we design a **multi-level attention** mechanism in order to discriminate the importance of different contexts and that of their features. In addition, we propose a **supervised attention mechanism** to align contextual features with those in the target review, so that the selected features for explanation can match to the user's preference.

5.1 Model Basics

As shown in Figure 3, the interaction module and the attention module are integrated into our model, so we first briefly introduce these two basic building blocks.

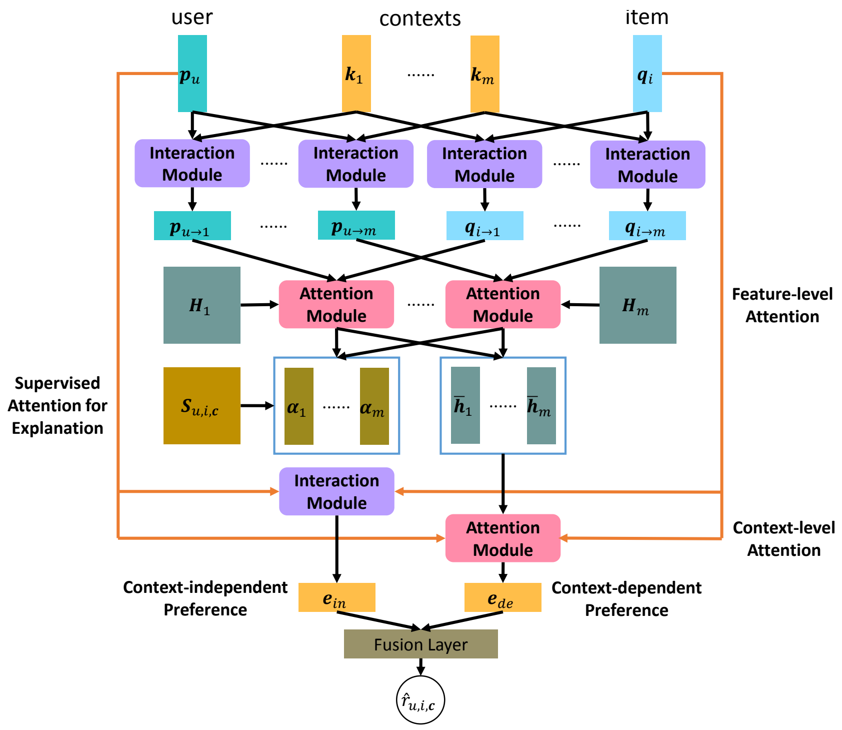


Fig. 3 Overview of our proposed model CAESAR, where $\mathbf{p}_u$, $\mathbf{k}_1$, $\mathbf{k}_m$, $\mathbf{q}_i$, $\mathbf{H}_1$, $\mathbf{H}_m$, and $\mathbf{S}_{u,i,\mathbf{c}}$ are input embeddings, and $\hat{r}_{u,i,\mathbf{c}}$ is the final predicted rating. $\mathbf{S}_{u,i,\mathbf{c}}$ corresponds to the ground-truth distribution of context-aware features in the user review.

Interaction Module. We propose to employ multi-layer perceptron [31,25,12] to learn the non-linear interaction between two entities among users, items, and contexts, since it has been demonstrated with better representation ability [16] than linear interaction such as matrix factorization [32]. For the convenience of later use, we write this module as:

$$\mathbf{v}_{a\to b} = Inter(\mathbf{v}_a, \mathbf{v}_b) \tag{3}$$

where $\mathbf{v}_a \in \mathbb{R}^d$ and $\mathbf{v}_b \in \mathbb{R}^d$ are input vectors, $Inter(\cdot)$ denotes the function of interaction module, and $\mathbf{v}_{a\to b} \in \mathbb{R}^d$ is the output vector. As shown in Figure 4, we first map the embeddings of two entities $\mathbf{v}_a$ and $\mathbf{v}_b$ to a shared hidden space via a bilinear layer. Then the resultant vector is fed into a stack of fully connected layers to enable non-linear interactions. Finally, the output vector from the last hidden layer is transformed into $\mathbf{v}_{a\to b}$, so that it has the same dimension as the input vectors. More specifically, the interaction module in our model is formally defined as:

$$\begin{aligned}
\mathbf{z}_0 &= \sigma(\mathbf{W}_0[\mathbf{v}_a, \mathbf{v}_b] + \mathbf{b}_0) \\
\mathbf{z}_1 &= \sigma(\mathbf{W}_1\mathbf{z}_0 + \mathbf{b}_1) \\
&\dots\dots \\
\mathbf{z}_L &= \sigma(\mathbf{W}_L\mathbf{z}_{L-1} + \mathbf{b}_L) \\
\mathbf{v}_{a\to b} &= \mathbf{W}_{L+1}\mathbf{z}_L + \mathbf{b}_{L+1}
\end{aligned} \tag{4}$$

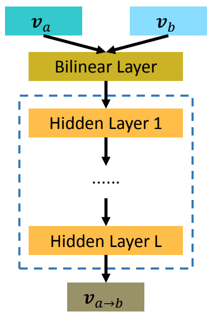


Fig. 4 Interaction module.

where $[\cdot, \cdot]$ denotes the concatenation of vectors, $\sigma(\cdot)$ is a non-linear activation function, $\mathbf{W}_x \in \mathbb{R}^{2d\times 2d}$ and $\mathbf{b}_x \in \mathbb{R}^{2d}$ are respectively the weight matrix and bias vector in the hidden layers, and $\mathbf{W}_{L+1} \in \mathbb{R}^{d\times 2d}$ and $\mathbf{b}_{L+1} \in \mathbb{R}^{d}$ correspond to the weight and bias in the final linear layer.

Attention Module. The attention mechanism [36,6,8,25,29,43] is employed on contexts and contextual features, because users under different contextual situations may have different needs, and different contexts may have different impacts. We formally denote the attention module as:

$$\overline{\mathbf{v}}, \boldsymbol{\alpha} = Attn(\mathbf{V}, \mathbf{v}_a, \mathbf{v}_b) \tag{5}$$

where $\mathbf{v}_a \in \mathbb{R}^d$ and $\mathbf{v}_b \in \mathbb{R}^d$ are the input vectors, $\mathbf{V} \in \mathbb{R}^{d\times z}$ represents the input matrix, $Attn(\cdot)$ denotes the function of attention module, $\overline{\mathbf{v}} \in \mathbb{R}^d$ is the aggregated output vector, and $\boldsymbol{\alpha} \in \mathbb{R}^z$ is the output vector consisting of attention scores. As illustrated in Figure 5, each object $\mathbf{v}_l \in \mathbf{V}$ is fed into the attention network together with input vectors $\mathbf{v}_a$ and $\mathbf{v}_b$ for computing a weight score. Formally, we define the attention network as:

$$\begin{aligned} \alpha_l^* &= \mathbf{w}^T \sigma(\mathbf{W}[\mathbf{v}_l, \mathbf{v}_a, \mathbf{v}_b] + \mathbf{b}) + b \\ \alpha_l &= \frac{\exp(\alpha_l^*)}{\sum_{l'=1}^{z} \exp(\alpha_{l'}^*)} \end{aligned} \tag{6}$$

where $\mathbf{W} \in \mathbb{R}^{3d\times 3d}$, $\mathbf{b} \in \mathbb{R}^{3d}$, $\mathbf{w} \in \mathbb{R}^{3d}$, and $b \in \mathbb{R}$ are model parameters. $\boldsymbol{\alpha}^*$ is normalized through the softmax function, resulting in the final attention vector $\boldsymbol{\alpha}$, with which we calculate the weighted sum $\overline{\mathbf{v}} = \sum_{l'=1}^{z} \alpha_{l'} \mathbf{v}_{l'}$ over the input embedding matrix $\mathbf{V}$.

5.2 Personalized Recommendation

As shown in Figure 3, given a user u, an item i, and the contextual combination $\mathbf{c}$, we first retrieve their corresponding embeddings $\mathbf{p}_u$, $\mathbf{q}_i$, and $\mathbf{k}_j$ for each context c_j in $\mathbf{c}$, where $j \in \{1, 2, ..., m\}$. To model the effects of contexts on a user, we employ the interaction module on the embeddings of user u and each context c_j via Equation (3), $\mathbf{p}_{u\to j} = Inter(\mathbf{p}_u, \mathbf{k}_j)$. In a similar way, we obtain the item's contextual embedding $\mathbf{q}_{i\to j}$.

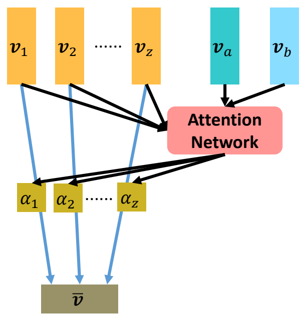


Fig. 5 Attention module.

As the contextual embeddings $\mathbf{p}_{u\to j}$ and $\mathbf{q}_{i\to j}$ are latent, we propose to characterize users' preferences over contextual features. For this purpose, with the contextual embeddings $\mathbf{p}_{u\to j}$ and $\mathbf{q}_{i\to j}$, and the corresponding embedding matrix of contextual features $\mathbf{H}_j$ for context c_j, we obtain the overall assessment of all the features $\overline{\mathbf{h}}_j$ and their attention scores $\boldsymbol{\alpha}_j$ using Equation (5), i.e., $\overline{\mathbf{h}}_j, \boldsymbol{\alpha}_j = Attn(\mathbf{H}_j, \mathbf{p}_{u\to j}, \mathbf{q}_{i\to j})$. Notice that we apply different attention modules to contexts in $\mathbf{c}$, and call them **feature-level attention** as a whole. In this way, different contextual features' importance to a certain context can be individually learned.

To further distinguish the importance of different contexts to the target user and item, we represent the embeddings from feature-level attention component as a matrix:

$$\overline{\mathbf{H}} = \begin{bmatrix} | & & | & & | \\ \overline{\mathbf{h}}_1 & \dots & \overline{\mathbf{h}}_j & \dots & \overline{\mathbf{h}}_m \\ | & & | & & | \end{bmatrix}$$

where $\overline{\mathbf{H}} \in \mathbb{R}^{d\times m}$ and $\overline{\mathbf{h}}_j \in \mathbb{R}^d$. Then, the influences of different contexts can be modeled via another attention module, i.e., $\mathbf{e}_{de}, \boldsymbol{\beta} = Attn(\overline{\mathbf{H}}, \mathbf{p}_u, \mathbf{q}_i)$, which we call **context-level attention**. Here $\mathbf{e}_{de}$ is *context-dependent preference* because it is inferred from contextual data including contexts and contextual features.

On the other hand, there may exist another type of preference, i.e., *context-independent preference*, which remains stable regardless of the context [7]. To model it, we employ another interaction module without contexts involved, i.e., $\mathbf{e}_{in} = Inter(\mathbf{p}_u, \mathbf{q}_i)$.

We finally pass both types of preference into a fusion layer, by which the rating can be predicted as:

$$\hat{r}_{u,i,\mathbf{c}} = \mathbf{w}_r^T[\mathbf{e}_{in}, \mathbf{e}_{de}] + b_r \tag{7}$$

where $\mathbf{w}_r \in \mathbb{R}^{2d}$ and $b_r \in \mathbb{R}$. For the rating prediction task, we adopt the mean squared error loss as its objective function:

$$\mathcal{L}_r = \sum_{u,i,\mathbf{c}\in\mathcal{T}} (r_{u,i,\mathbf{c}} - \hat{r}_{u,i,\mathbf{c}})^2 \tag{8}$$

where $\mathcal{T}$ is the training set and $r_{u,i,\mathbf{c}}$ denotes the ground truth rating that the user u assigned to item i under contexts $\mathbf{c}$.

5.3 Explanation Generation

The embedding matrix $\mathbf{H}_j$ leverages contextual features in the feature-level attention component, but they are latent because there is no direct connection with explicit features. To build such connection for generating explanation, we force the scores resulting from feature-level attention to be the same as the ground-truth frequencies of contextual features in the target review.

Concretely, from the target review we can have the occurrence frequency $freq_f^{c_j}$ of each feature f that appears in context c_j's feature collection $\mathcal{F}_j^{c_j}$ (from Section 4). We can then obtain the feature distribution vector $\mathbf{s}_j$ (the matrix form is $\mathbf{S}_{u,i,\mathbf{c}}$ as shown in Figure 3), in which each element corresponds to a contextual feature's normalized frequency:

$$s_j^f = freq_f^{c_j} / \sum_{f' \in \mathcal{F}_j^{c_j}} freq_{f'}^{c_j} \tag{9}$$

To align the attention vector $\boldsymbol{\alpha}_j$ with the distribution vector $\mathbf{s}_j$, we apply the mean squared error loss function:

$$\mathcal{L}_e = \sum_{u,i,\mathbf{c} \in \mathcal{T}} \sum_{j=1}^{m} \sum_{k=1}^{n} (s_j^k - \alpha_j^k)^2 \tag{10}$$

where m is the total number of contexts in $\mathbf{c}$ and n denotes the number of contextual features. As the attention scores are learned from explicit contextual features through supervised learning, we name it **supervised attention**. To explain the recommendation, we select a context with the highest score from context-level attention and its most relevant contextual features from feature-level attention to fill in the template: "*This product is recommended to you, because its [features] are suitable for your current [context].*"

5.4 Multi-task Learning

At last, we integrate the rating prediction task and the context-aware explanation task into a unified multi-task learning framework, for which the objective function is:

$$\mathcal{J} = \min_{\Theta}(\lambda_r \mathcal{L}_r + \lambda_e \mathcal{L}_e + \lambda_n ||\Theta||^2) \tag{11}$$

where Θ is the set of model parameters, and λ_r, λ_e and λ_n are regularization weights for different tasks.

6 Experiment on rating prediction

In this section, we present our experimental results in comparison with baseline models. We also study some important hyper-parameters of our model CAESAR.

Table 2 Statistics of our datasets

	TripAdvisor	**Yelp-2019**
# of users	9,765	27,147
# of items	6,280	20,266
# of reviews	320,023	1,293,247
Avg. # of reviews per user	32.77	47.64
Avg. # of reviews per item	50.96	63.81
# of variables[a] in *companion*	6	-
# of variables in *day of a week*	-	7
# of variables in *month*	13	12
# of variables in *destination*	415	242

[a] *Variable* represents *contextual variable.*

6.1 Datasets

In our experiment, we used two large-scale datasets from two typical service domains, i.e., hotel and restaurant, to evaluate our proposed model. For hotel domain, we constructed the dataset with reviews collected from a popular travel website TripAdvisor[1]. Specifically, we crawled all the user reviews to every hotel located in Hong Kong from this website, as well as those users' historical reviews in other cities. After removing non-English reviews, we have in total 2,118,108 interaction records. The other dataset is from Yelp Challenge 2019[2], which contains 6,685,900 restaurant reviews. Since the two datasets are very large while we do not focus on handling the cold start problem in this paper, we further processed them by recursively removing users and items with less than 20 interactions, which results in two subsets TripAdvisor and Yelp-2019 (see Table 2 for their descriptive characteristics).

Each review record in our datasets comprises user ID, item ID, overall rating in the scale of 1 to 5, textual review, and contexts in which the user was experiencing that item. As indicated in [31], taking more contextual categories into account may lead to better performance, so we make use of all the available contexts associated with each review. Specifically, the contextual categories consist of *companion*, *month*, and *destination* for TripAdvisor dataset, and *day of a week*, *month*, and *destination* for Yelp-2019. Notice that the contextual categories *companion* and *month* in TripAdvisor indicate with whom and in which month a user stayed in a hotel, but Yelp-2019 does not provide the exact time a user was dining in a restaurant, so the contextual variables for *day of a week* and *month* are inferred from the review time. For the contextual category *destination* in both datasets, we took the target city (where the item is located) as the contextual variable.

6.2 Compared Methods

To evaluate the performance of our CAESAR, we compare it with the following state-of-the-art methods:

[1] https://www.tripadvisor.com
[2] https://www.yelp.com/dataset/challenge

- **PMF**: Probabilistic Matrix Factorization [32]. This is the standard matrix factorization method that characterizes users and items by latent factors inferred from observed ratings. We take it as the **context-unaware baseline**. Its objective function is optimized by alternative least square (ALS).
- **EFM**: Explicit Factor Models [46]. It is a joint matrix factorization model in which user-feature attention and item-feature quality are considered for explaining recommendations. It is the **feature-level explanation baseline**, without considering user contexts. Stochastic gradient descent (SGD) is introduced to optimize its objective function in our implementation.
- **AFM**: Attentional Factorization Machines [43]. This model extends factorization machines (FM) [33] by learning the importance of each feature interaction via a neural attention network. It is a context-aware approach, but treats users, items, and contexts equally as sparse features.
- **NFM**: Neural Factorization Machines [15]. It is a more generalized FM built upon neural network for learning high-order feature interactions in a non-linear way for dealing with sparse data. The way it models contexts is similar to AFM.
- **AIN**: Attentive Interaction Network [31]. To model the effects of contexts on users and items, this model employs two pathways of interaction module, followed by attention mechanism to discriminate the impacts of different contexts on users and items.

We omit the comparison with other context-aware models such as Multiverse [19], FM [33], CAMF [3] and COT [28], since they are outperformed by the recently proposed AIN as shown in [31].

6.3 Evaluation Metrics

To compare the recommendation performance of different methods, we adopt two commonly used metrics in recommender systems: Root Mean Square Error (RMSE) and Mean Absolute Error (MAE).

RMSE is calculated by estimating the quadratic difference between ground-truth rating $r_{u,i,\mathbf{c}}$ and the predicted one $\hat{r}_{u,i,\mathbf{c}}$:

$$RMSE = \sqrt{\frac{1}{N}\sum_{u,i,\mathbf{c}}(r_{u,i,\mathbf{c}} - \hat{r}_{u,i,\mathbf{c}})^2} \tag{12}$$

where N is the number of instances in the testing set.

Similarly, MAE can be calculated via the following formula:

$$MAE = \frac{1}{N}\sum_{u,i,\mathbf{c}}|r_{u,i,\mathbf{c}} - \hat{r}_{u,i,\mathbf{c}}| \tag{13}$$

For both metrics, a lower value indicates a better performance.

6.4 Experimental Settings

We randomly divide each dataset into training (80%), validation (10%), and testing (10%) sets, and guarantee that each user/item has at least one instance in the

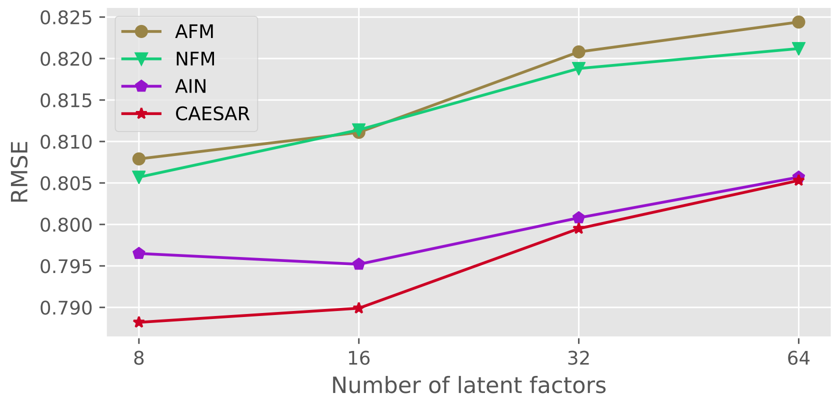


Fig. 6 RMSE on TripAdvisor with varied numbers of latent factors in four compared context-aware methods.

training set. We repeat the splitting process for 5 times. The validation set is used for hyper-parameters tuning, and we report the average performance on the testing set. The early stopping strategy is performed on all the models, i.e., a model's RMSE and MAE on the testing set are reported when it reaches the best performance on the validation set.

We implemented AIN and our CAESAR in Python[3] using TensorFlow[4], and adopted the codes of AFM and NFM shared by their authors. All the neural network based methods, i.e., AFM, NFM, AIN and CAESAR, are optimized by Adam [22]. The initial learning rate of AFM and NFM is searched in [0.005, 0.01, 0.02, 0.05], following the setting in [15]. The number of latent factors of these models is searched in [8, 16, 32, 64]. Figure 6 shows the recommendation accuracy by varying the value of this parameter on TripAdvisor dataset[5]. From the figure, we can see that these methods generally share the similar trend: Fewer latent factors (e.g., 8 for AFM, NFM and CAESAR, and 16 for AIN) lead to better accuracy, while increasing the number of latent factors deteriorates the performance, because too many latent factors may cause overfitting. Therefore, on TripAdvisor we set the number of latent factors of AFM, NFM and CAESAR to 8, and that of AIN to 16. Matrix factorization methods PMF and EFM were also implemented in Python. For PMF, we set the dimension of latent factors to 20 following [20], and search trade-off parameters in [0.1, 1, 10, 100]. For EFM, the numbers of explicit factors and implicit factors are set equally and both searched in [8, 16, 32, 64]. We reuse the other hyper-parameters of the baselines as reported in the original papers. Furthermore, the weight and bias parameters of all the methods are learned from scratch with random initialization, for the fair comparison.

For our model CAESAR, the learning rate is set to 0.0001, the batch size is 64, and the number of contextual features n for each context is 100. We use $ReLU(\cdot)$ as the activation function, and employ 2 hidden layers for interaction module in the condition of modeling context-independent preference and 1 hidden layer for

[3] https://www.python.org
[4] https://www.tensorflow.org
[5] The results on Yelp-2019 dataset are similar, so we do not show here.

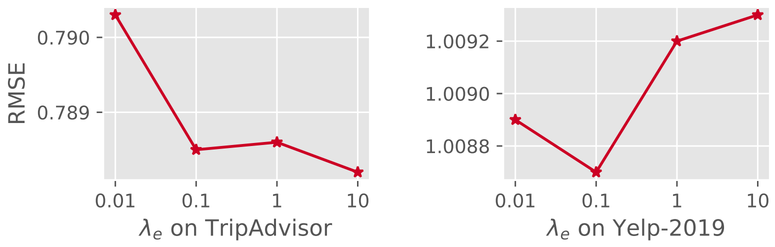


Fig. 7 RMSE on TripAdvisor (left) and Yelp-2019 (right) with different regularization weights on λ_e in our model CAESAR.

context-dependent preference. For the regularization weights on different tasks, we set λ_r to 1 and λ_n to 0.0001, and search λ_e in [0.01, 0.1, 1, 10]. The left part of Figure 7 shows our method's performance w.r.t. λ_e on TripAdvisor dataset and the right part shows that on Yelp-2019. As it can be seen, the optimal values of λ_e on two datasets are different, i.e., 10 for TripAdvisor and 0.1 for Yelp-2019. The larger the regularization weight λ_e is, the closer the scores from feature-level attention are to the distribution of contextual features in user reviews. Notice that since the contextual variables for *day of a week* and *month* in Yelp-2019 are not the exact time when a user was dining, the inferred features may not precisely match those expressed in the target review, which may cause difference between the attention scores and the distribution vector of contextual features (i.e., small λ_e).

6.5 Performance Comparison

The accuracy of our model CAESAR in comparison with baseline models on two datasets is given in Table 3. From this table, we can have several observations.

Firstly, CAESAR consistently achieves the best performance w.r.t. RMSE and MAE on two datasets. We attribute this to the fact that our model considers explicit contextual features extracted from users reviews as important information source, while features in the compared method like AIN are latent. This illustrates the necessity of fusing explicit features into the context-aware model, as well as demonstrating the capability of our proposed supervised attention mechanism to deal with these features. In addition, we notice that the performance gap between CAESAR and AIN is relatively small. Our primary goal has been to achieve explainability for which we proposed the supervised attention mechanism, which aligns the attention weights with the feature distribution in the target user review, but this mechanism that differs from vanilla attention may sacrifice accuracy to a certain degree.

Secondly, factorization machines (FM) based models (i.e., AFM and NFM) obtain similar performance, but are dominated by the other context-aware methods on two datasets. Although AFM and NFM enhance FM via neural network, they treat users, items and contexts as sparse features and model all types of interactions in the similar way. In comparison, the other context-aware methods (i.e., AIN and CAESAR) employ two pathways of interaction module to characterize

Table 3 Performance comparison of all methods in terms of RMSE and MAE. The best performing method and values are boldfaced. * indicates that our model CAESAR performs significantly better than NFM through Student's t-test ($p < 0.01$).

Model	TripAdvisor		Yelp-2019	
	RMSE	MAE	RMSE	MAE
PMF	0.8703	0.6961	1.0856	0.8765
EFM	0.8415	0.6403	1.0557	0.8243
AFM	0.8102	0.6181	1.0355	0.8060
NFM	0.8095	0.6177	1.0366	0.8056
AIN	0.7952	0.6072	1.0111	0.7879
CAESAR	**0.7898***	**0.6022***	**1.0080***	**0.7814***

the effects of different contexts on users and items, and subsequently adopt attention mechanism to dynamically infer representations of users and items, which may lead to better performance. This verifies the importance of modeling different types of interactions in different ways, and the respective roles of interaction module and attention module in our framework.

Thirdly, context-unaware models, i.e, PMF and EFM, underperform all the other methods that take into account contextual information during recommendation process. This is not surprising, as users are likely to make decision according to their contextual situations especially for service products. As such, context-aware models can better characterize users' preferences over item features. Besides, since PMF and EFM are both based on matrix factorization, the linearity characteristic may make user and item modeling limited [16], in comparison with other neural network based methods that model users and items in a non-linear way. In addition, we observe that the performance of EFM is much better than that of PMF, which may be because the former takes features extracted from user reviews as complementary information into its modeling process. This again demonstrates the usefulness of explicit features to improve recommendation performance.

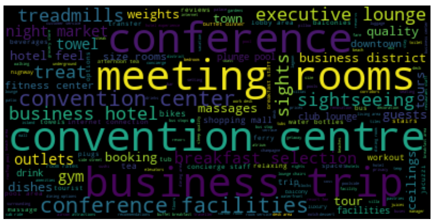
7 Explanation Study

To study our model's explainability, we provide analysis on features mined by our contextual feature mining approach and human evaluation on explanations as generated by CAESAR.

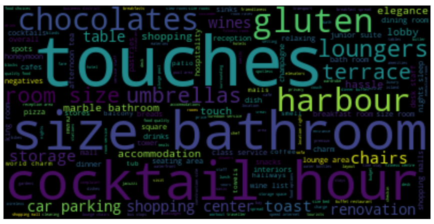
7.1 Contextual Feature Analysis

We generate Word Clouds for features on TripAdvisor dataset (see Figure 8), as mined by our contextual feature mining approach (the results on Yelp-2019 dataset show similar pattern). The size of features in sub-figures (a), (b) and (c) indicates the weight score computed via Equation (2), and that in (d) represents a feature's occurring frequency in all the user reviews.

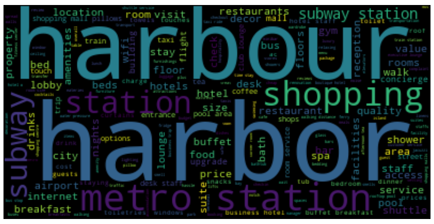
It can be seen that, under the contextual situation <u>*business*</u>, users have paid more attention to facilities like "*meeting rooms*", "*conference*" and "*convention centre*". Relatively, users under <u>*couples*</u> have preferred leisurely and romantic features, such as "*cocktail hour*" and "*chocolates*". For the contextual category *des-*



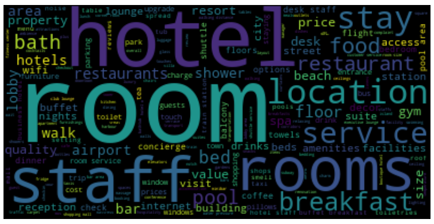
(a) Contextual features for <u>*business*</u>



(b) Contextual features for <u>*couples*</u>



(c) Contextual features for <u>*Hong Kong*</u>



(d) Features according to occurring frequency

Fig. 8 Word Clouds of features identified by our contextual feature mining approach (a, b, and c) and that based on occurring frequency (d) on TripAdvisor dataset.

tination, our approach is able to find features that reveal a place's characteristics, e.g., "*harbour*", "*shopping*", and "*metro station*" in <u>*Hong Kong*</u>. It hence shows that our contextual feature mining approach is capable of discovering context-aware features. In comparison, the features identified according to occurring frequency (Figure 8 (d), which have been commonly adopted by existing explainable recommendation approaches [46,39]) primarily reflect some general aspects such as "*hotel*", "*room*" and "*staff*", which are not context-specific.

To qualitatively measure the explanation features produced by different methods, we further present two automatic metrics, Accuracy and Diversity, on which our method CAESAR and EFM [46] (that produces context-unaware explanation) are compared. For each record in the testing set, we select the top 5 features for evaluation.

The intuition behind Accuracy is that the selected features for explaining a recommendation are expected to be as close to the target user's preferred ones as possible, so that the explanation could better justify the recommendation that caters to his/her needs. Therefore, it is formally defined as estimating the intersection of ground-truth features $\mathcal{F}_{u,i}$ and predicted ones $\hat{\mathcal{F}}_{u,i}$:

$$ACC = \frac{1}{N} \sum_{u,i} \left| \mathcal{F}_{u,i} \cap \hat{\mathcal{F}}_{u,i} \right| \tag{14}$$

where N is the number of instances in the testing set. For this metric, a larger value indicates a better performance.

Meanwhile, in our experimental trial, we notice that feature lists resulting from EFM are sometimes similar or even identical to each other, e.g., "*hotel*" and "*room*" are always contained due to their high occurring frequency. Explanations are expected to reflect the characteristics of the recommendations, and at the same time be personalized to users' preferences, but it may be difficult for identical

Table 4 Comparison between CAESAR and EFM in terms of feature accuracy (ACC) and diversity (DIV).

Model	TripAdvisor		Yelp-2019	
	ACC	DIV	ACC	DIV
EFM	1.3536	4.7398	0.3439	3.9655
CAESAR	0.5556	**1.4434**	0.0113	**0.7188**

explanations to achieve such goals. Besides, poor explanations may cause negative effect on users [37]. Hence, we propose Diversity to measure predicted features from this perspective. It measures the intersection between any pair of two predicted feature lists $\hat{\mathcal{F}}_{u,i}$ and $\hat{\mathcal{F}}_{u',i'}$:

$$DIV = \frac{2}{N \times (N-1)} \sum_{u,u',i,i'} \left| \hat{\mathcal{F}}_{u,i} \cap \hat{\mathcal{F}}_{u',i'} \right| \tag{15}$$

The lower, the better it is for this metric.

Table 4 shows the results on two datasets, from which we can see, although the Accuracy of EFM is better than that of our model, the former's Diversity is much worse. Since EFM selects features that frequently appear in user reviews for explanation, it may result in similar feature lists (as shown in examples of Table 5), so its diversity can be limited. Relatively, the features from our CAESAR are context-specific, so for different contexts, the features are likely different (see Figure 8 and Table 5), which to some extent reflects the generated explanation's personalization ability. However, the results on the two metrics may not be able to fully reflect the explainability of explanations, e.g., user acceptance. Moreover, the context-aware explainability are not evaluated by the two metrics. To more comprehensively evaluate explanations, we conduct a small-scale human evaluation in the following section.

7.2 Human Evaluation on Explainability

To evaluate the quality of explanations produced by our method CAESAR, we conduct a human evaluation on TripAdvisor dataset, because its contexts are more precise than that of Yelp-2019 as discussed in Section 6.4.

In our questionnaire, we prepared two questions, each containing 10 different cases. We then invited 10 people to answer those questions. Specifically, they all are Chinese, and currently doing either Ph.D. or M.Phil. in computer science, so their English language proficiency is qualified for this evaluation. Their gender distribution is well balanced (5 males vs. 5 females), and their ages are distributed between 23 and 30. As a consequence of the demographics, their evaluation may only reflect a small group of people's perception to the explanations. For example, American with high school diploma may value the explanations differently. Moreover, since the participants are experts in the field of computer science, their domain knowledge may cause certain bias to the evaluation. For example, it could be easier for them than ordinary people to understand what "*context*" means, so their evaluation may be biased to more detailed explanations, which, however, may not fully reflect the real world situation. In order to quickly obtain some initial

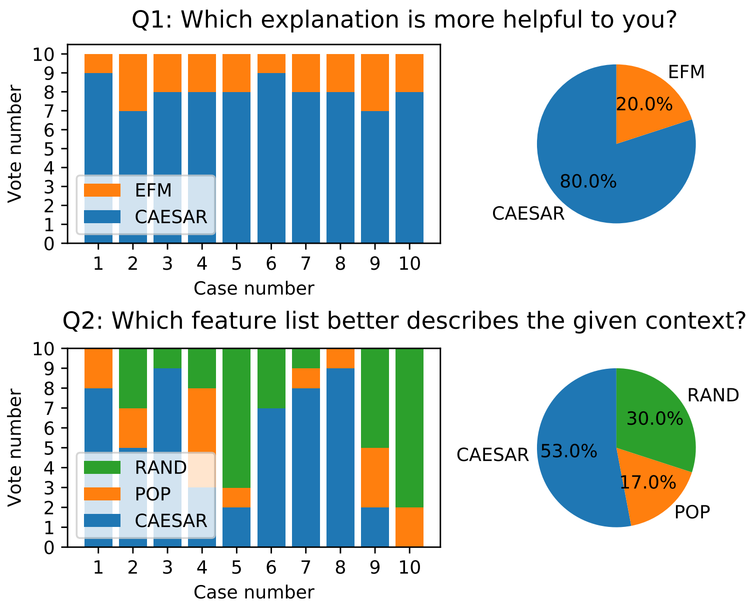


Fig. 9 Results of human evaluation on explanations provided by compared methods on TripAdvisor dataset. The bar charts show the exact vote numbers on different testing cases, and the pie charts illustrate the voting percentages.

results, in this paper we conduct a small-scale human evaluation, and leave the large-scale one for future work.

The first question (Q1) aims at studying which type of explanations, either context-aware or context-unaware, could be more helpful. For each case we provided two explanations respectively returned by our method CAESAR and the baseline EFM [46]. Specifically, the top 5 features as selected by each method are filled in its explanation template as shown in Table 5. In addition, our method adds one context related to those features. Along with the explanation and its corresponding recommendation (a hotel), we also provided the imagined user's contexts and her/his review to the hotel. The human judges were then asked to indicate which explanation is more helpful for them to understand the recommendation.

The second question (Q2) was designed to investigate whether the features given in the explanation could well describe the selected context. Considering the fact that there is no available context-aware explanation method to compare at the time of our experiment, we adopted two simple baselines: RAND and POP. In more details, given a context and its context-aware features obtained from Section 4, RAND randomly selects some context-aware features, while POP selects features based on its occurring frequency. Notice that, the details of those three compared methods were hidden to human judges and the feature lists were randomly shuffled, which was to fairly compare their performance. Then, the human judges were asked to evaluate which feature list is more suitable to describe the given contextual situation.

Specifically, the two evaluation questions, each with one example case, are displayed below.

- **Q1**: After reading the given information, do you think which explanation is more helpful for you to understand the recommendation?

  Hotel: Novotel Citygate Hong Kong
  Context: travelled to Hong Kong as a Couple on February
  Review: Stayed March 2013 between flights from China mainland and homebound. Hotel is a short distance from the airport via free shuttle every 15 mins. The hotel is located in the small residential area of Tung Chung which is a very modern and attractive place to walk around. There is quite a bit of shopping near the hotel including clothing label outlet stores and local shopping/supermarket etc. There is an MTR station and bus stop at the base of the Citigate Mall and the Ferry and Cable Car is a short walk from the hotel. Disneyland is just up the road and the Giant Buddha at the top of the hill (Cable Car, bus or taxi). So the Citigate offers more touring options than a typical airport hotel. The hotel meets the typical Accor expectations and honors the various loyalty cards (including Advantage Plus) for both accommodation and discount dining. It is not as flashy as some Hong Kong hotels but is true to Western expectations of the Accor brand. We enjoyed our stay and recommend the hotel if you want/need something close to the airport. Breakfast buffet is good but not excellent.

  - A. This hotel is recommended to you, because its features [symphony, mtr station, mongkok, bldg, airport access] are suitable for your current context [Hong Kong].
  - B. You might be interested in features [room, hotel, staff, rooms, location], on which this hotel performs well.

- **Q2**: Imagine you are under a specific context when booking a hotel, which feature list do you think better describes this context?

  Context: December

  - A. floor, selections, apartment, queue, service
  - B. christmas market, premier rooms, birthday stay, flyover, island side
  - C. service, floor, area, walk, property

We keep EFM's original template format as it has been widely adopted in related work [13,39,46], and design our own one which is more suitable for context-aware explanation. The template difference could potentially influence the response of participants, but we believe that the highlighted features/contexts are what they concern. The evaluation results are depicted in Figure 9, where the bar charts show the vote numbers on the 10 testing cases for different methods, and the pie charts show the percentages of votes.

For Q1, regarding all of the cases, most human judges regard the explanations generated by our method CAESAR being more helpful for them to understand the recommendation. This validates our model's capability of returning more useful explanations. As to Q2, our method obtains 53% votes, which is obviously higher than those of the two baselines (30% to RAND and 17% to POP). This confirms that to a large extent our method can find features that better characterize a context. From the bar chart, we also observe that our method gains more than 5 votes on 6 different cases, while RAND dominates on 3 cases, leaving POP on only one case. This might be because RAND and POP both leverage the context-aware features as obtained through our feature mining approach (see Section 4), which may enable them to return some reasonable features in some cases. This again demonstrates the effectiveness of our contextual feature mining approach in finding context-aware features.

7.3 Case Study on Explanations

To qualitatively compare the explanations generated by CAESAR and those by EFM [46], a case study is shown in Table 5. The context and feature highlighted in green are our model's selected information elements to generate the context-aware explanation; while for EFM only the selected feature is highlighted because it is context-unaware.

From the examples we can see that when the same user visits two different hotels under different contexts, our model can adaptively select the most important context to her as well as the most relevant features. More specifically, our model selects "*rooftop view*", "*floor levels*", "*smorgasbord*", and "*eating establishments*", to match the *couples* context, and selects "*harbor*" and "*location*" for the destination *Sydney*. In comparison, EFM just selects some general features for the two hotels, such as "*room*" and "*staff*", which can not distinguish the user's needs under different contexts.

8 Conclusion

In this paper, we propose a novel neural model called CAESAR, which is experimentally demonstrated to be capable of characterizing users' preferences over contextual features mined from user reviews, for achieving both context-aware recommendation and explanation. In particular, our designed multi-level attention mechanism can distinguish the importance of different contexts and their related features. Moreover, the supervised attention can align explicit contextual features with implicit ones to generate context-aware explanation, while still being able to enhance recommendation accuracy simultaneously. Experimental results on two real-world datasets (TripAdvisor and Yelp) show that our model outperforms the state-of-the-art baselines in terms of not only rating prediction accuracy, but also returning context-aware feature-level explanations that are more helpful than context-unaware explanations as judged by human evaluators.

In the future, we plan to obtain more negative features and integrate them into user preference modeling, so as to highlight both *pros* and *cons* of the recommended item through explanation. We will also further strengthen our model's recommendation and explanation performance by pre-training the embeddings of contextual features. In addition, we are interested in modeling real-time changing contexts that can be critical in the mobile environment, e.g., finding nearby restaurants.

Table 5 A case study for explanation comparison between our method CAESAR and a baseline EFM when recommending two hotels to the same user. The texts highlighted in green are automatically selected context/feature to fill in the explanation template.

Item	Contexts		Features		*Features in testing review*	**Explanation**	
	CAESAR	EFM	CAESAR	EFM		CAESAR	EFM
Hotel Madera Hong Kong	Couples, September, Hong Kong	-	rooftop view, parking bays, floor levels, smorgasbord, eating establishments	room, hotel, staff, stay	bar, rooftop view, hotels, cafe, rooftop, bedroom, balcony, breakfast, location, glass, floors, room, station, hotel	*This product is recommended to you, because its [**rooftop view**] is suitable for your current context [**couples**].*	*You might be interested in [**room**], on which this product performs well.*
Meriton Suites Kent Street	Business, October, Sydney	-	harbor, location, apartment, lifts	room, hotel, staff, location	location, windows, noise, rooms	*This product is recommended to you, because its [**harbor**] is suitable for your current context [**Sydney**].*	*You might be interested in [**room**], on which this product performs well.*

Acknowledgements This is a post-peer-review, pre-copyedit version of an article published in Journal of Intelligent Information Systems (JIIS). The final authenticated version is available online at: https://doi.org/10.1007/s10844-020-00631-8

References

1. Abowd, G.D., Dey, A.K., Brown, P.J., Davies, N., Smith, M., Steggles, P.: Towards a better understanding of context and context-awareness. In: International symposium on handheld and ubiquitous computing, pp. 304–307. Springer (1999)
2. Adomavicius, G., Tuzhilin, A.: Context-aware recommender systems. In: B. Shapira (ed.) Recommender Systems Handbook, 2 edn., chap. 6, pp. 191–226. Springer (2015)
3. Baltrunas, L., Ludwig, B., Ricci, F.: Matrix factorization techniques for context aware recommendation. In: Proceedings of the fifth ACM conference on Recommender systems, pp. 301–304 (2011)
4. Baral, R., Zhu, X., Iyengar, S., Li, T.: Reel: Review aware explanation of location recommendation. In: Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization, pp. 23–32. ACM (2018)
5. Catherine, R., Cohen, W.: Transnets: Learning to transform for recommendation. In: Proceedings of the Eleventh ACM Conference on Recommender Systems, pp. 288–296. ACM (2017)
6. Chen, C., Zhang, M., Liu, Y., Ma, S.: Neural attentional rating regression with review-level explanations. In: Proceedings of the 2018 World Wide Web Conference on World Wide Web, pp. 1583–1592. International World Wide Web Conferences Steering Committee (2018)
7. Chen, G., Chen, L.: Augmenting service recommender systems by incorporating contextual opinions from user reviews. User Modeling and User-Adapted Interaction **25**(3), 295–329 (2015)
8. Chen, J., Zhang, H., He, X., Nie, L., Liu, W., Chua, T.S.: Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention. In: Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval, pp. 335–344. ACM (2017)
9. Chen, X., Zhang, Y., Qin, Z.: Dynamic explainable recommendation based on neural attentive models. In: Thirty-Third AAAI Conference on Artificial Intelligence. AAAI Press (2019)
10. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 1724–1734 (2014)
11. Costa, F., Ouyang, S., Dolog, P., Lawlor, A.: Automatic generation of natural language explanations. In: Proceedings of the 23rd International Conference on Intelligent User Interfaces Companion, p. 57. ACM (2018)
12. Deng, Z.H., Huang, L., Wang, C.D., Lai, J.H., Yu, P.S.: Deepcf: A unified framework of representation learning and matching function learning in recommender system. In: Thirty-Third AAAI Conference on Artificial Intelligence. AAAI Press (2019)
13. Gao, J., Wang, X., Wang, Y., Xie, X.: Explainable recommendation through attentive multi-view learning. In: Thirty-Third AAAI Conference on Artificial Intelligence. AAAI Press (2019)
14. He, X., Chen, T., Kan, M.Y., Chen, X.: Trirank: Review-aware explainable recommendation by modeling aspects. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, pp. 1661–1670. ACM (2015)
15. He, X., Chua, T.S.: Neural factorization machines for sparse predictive analytics. In: Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval, pp. 355–364. ACM (2017)
16. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.S.: Neural collaborative filtering. In: Proceedings of the 26th International Conference on World Wide Web, pp. 173–182. International World Wide Web Conferences Steering Committee (2017)
17. Herlocker, J.L., Konstan, J.A., Riedl, J.: Explaining collaborative filtering recommendations. In: Proceedings of the 2000 ACM conference on Computer supported cooperative work, pp. 241–250. ACM (2000)

18. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
19. Karatzoglou, A., Amatriain, X., Baltrunas, L., Oliver, N.: Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In: Proceedings of the fourth ACM conference on Recommender systems, pp. 79–86. ACM (2010)
20. Kim, D., Park, C., Oh, J., Lee, S., Yu, H.: Convolutional matrix factorization for document context-aware recommendation. In: Proceedings of the 10th ACM Conference on Recommender Systems, pp. 233–240. ACM (2016)
21. Kim, Y.D., Choi, S.: Nonnegative tucker decomposition. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8. IEEE (2007)
22. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: International Conference on Learning Representations (ICLR) (2015)
23. Levi, A., Mokryn, O., Diot, C., Taft, N.: Finding a needle in a haystack of reviews: cold start context-based hotel recommender system. In: Proceedings of the sixth ACM conference on Recommender systems, pp. 115–122. ACM (2012)
24. Li, L., Chen, L., Zhang, Y.: Towards controllable explanation generation for recommender systems via neural template. In: Companion Proceedings of the Web Conference 2020, pp. 198–202 (2020)
25. Li, L., Dong, R., Chen, L.: Context-aware co-attention neural network for service recommendations. In: Proceedings of ICDE'19 Workshop on Recommender Systems with Big Data, pp. 201–208. IEEE (2019)
26. Li, L., Zhang, Y., Chen, L.: Generate neural template explanations for recommendation. In: Proceedings of the 29th ACM International Conference on Information and Knowledge Management, pp. 755–764. ACM (2020)
27. Li, P., Wang, Z., Ren, Z., Bing, L., Lam, W.: 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, pp. 345–354. ACM (2017)
28. Liu, Q., Wu, S., Wang, L., et al.: Cot: Contextual operating tensor for context-aware recommender systems. In: Twenty-ninth AAAI Conference on Artificial Intelligence. AAAI Press (2015)
29. Lu, Y., Dong, R., Smyth, B.: Coevolutionary recommendation model: Mutual learning between ratings and reviews. In: Proceedings of the 2018 World Wide Web Conference on World Wide Web, pp. 773–782. International World Wide Web Conferences Steering Committee (2018)
30. Lu, Y., Dong, R., Smyth, B.: Why i like it: Multi-task learning for recommendation and explanation. In: Proceedings of the Eighth ACM Conference on Recommender Systems. ACM (2018)
31. Mei, L., Ren, P., Chen, Z., Nie, L., Ma, J., Nie, J.Y.: An attentive interaction network for context-aware recommendations. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, pp. 157–166. ACM (2018)
32. Mnih, A., Salakhutdinov, R.R.: Probabilistic matrix factorization. In: Advances in neural information processing systems, pp. 1257–1264 (2008)
33. Rendle, S.: Factorization machines. In: 10th International Conference on Data Mining (ICDM), pp. 995–1000. IEEE (2010)
34. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th international conference on World Wide Web, pp. 285–295. ACM (2001)
35. Sato, M., Ahsan, B., Nagatani, K., Sonoda, T., Zhang, Q., Ohkuma, T.: Explaining recommendations using contexts. In: 23rd International Conference on Intelligent User Interfaces, pp. 659–664. ACM (2018)
36. Seo, S., Huang, J., Yang, H., Liu, Y.: Interpretable convolutional neural networks with dual local and global attention for review rating prediction. In: Proceedings of the Eleventh ACM Conference on Recommender Systems, pp. 297–305. ACM (2017)
37. Tintarev, N., Masthoff, J.: Explaining recommendations: Design and evaluation. In: B. Shapira (ed.) Recommender Systems Handbook, 2 edn., chap. 10, pp. 353–382. Springer (2015)
38. Vig, J., Sen, S., Riedl, J.: Tagsplanations: explaining recommendations using tags. In: Proceedings of the 14th international conference on Intelligent user interfaces, pp. 47–56 (2009)

39. Wang, N., Wang, H., Jia, Y., Yin, Y.: Explainable recommendation via multi-task learning in opinionated text data. In: Proceedings of the 41st international ACM SIGIR conference on Research & development in information retrieval, pp. 165–174. ACM (2018)
40. Wang, X., Chen, Y., Yang, J., Wu, L., Wu, Z., Xie, X.: A reinforcement learning framework for explainable recommendation. In: 2018 IEEE International Conference on Data Mining (ICDM), pp. 587–596. IEEE (2018)
41. Wang, X., He, X., Feng, F., Nie, L., Chua, T.S.: Tem: Tree-enhanced embedding model for explainable recommendation. In: Proceedings of the 2018 World Wide Web Conference on World Wide Web, pp. 1543–1552. International World Wide Web Conferences Steering Committee (2018)
42. Wang, X., Wang, D., Xu, C., He, X., Cao, Y., Chua, T.S.: Explainable reasoning over knowledge graphs for recommendation. In: Thirty-Third AAAI Conference on Artificial Intelligence. AAAI Press (2019)
43. Xiao, J., Ye, H., He, X., Zhang, H., Wu, F., Chua, T.S.: Attentional factorization machines: Learning the weight of feature interactions via attention networks. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (2017)
44. Xin, X., Chen, B., He, X., Wang, D., Ding, Y., Jose, J.: Cfm: convolutional factorization machines for context-aware recommendation. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence, pp. 3926–3932. AAAI Press (2019)
45. Zhang, Y., Chen, X.: Explainable recommendation: A survey and new perspectives. Foundations and Trends® in Information Retrieval **14**(1), 1–101 (2020)
46. Zhang, Y., Lai, G., Zhang, M., Zhang, Y., Liu, Y., Ma, S.: 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, pp. 83–92. ACM (2014)
47. Zhang, Y., Zhang, H., Zhang, M., Liu, Y., Ma, S.: Do users rate or review? boost phrase-level sentiment labeling with review-level sentiment classification. In: Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval, pp. 1027–1030. ACM (2014)
48. Zhao, L., Song, K., Sun, C., Zhang, Q., Huang, X., Liu, X.: Review response generation in e-commerce platforms with external product information. In: The World Wide Web Conference, pp. 2425–2435. ACM (2019)