Self-Attentive Recommendation for Multi-Source Review Package

Pin-Yu Chen
National Yang Ming Chiao Tung University
Hsinchu, Taiwan
pinyu.eed04@nctu.edu.tw

Yu-Hsiu Chen
National Yang Ming Chiao Tung University
Hsinchu, Taiwan
yhchen.cm06g@nctu.edu.tw

Hong-Han Shuai
National Yang Ming Chiao Tung University
Hsinchu, Taiwan
hhshuai@nctu.edu.tw

Yung-Ju Chang
National Yang Ming Chiao Tung University
Hsinchu, Taiwan
armuro@cs.nctu.edu.tw

***Abstract*—With the diversified sources satisfying users' needs, many online service platforms collect information from multiple sources in order to provide a set of useful information to the users. However, existing recommendation systems are mostly designed for single-source data, and thus fail to recommend multi-source review packages since the interplay between the reviews of different sources is not properly modeled. In fact, modeling the interplay between different sources is challenging because 1) two reviews may conflict with each other, 2) different users have different preferences on review sources, and 3) users' preferences to each source may change under different scenarios. To address these challenges, we propose Self-Attentive Recommendation for multi-source review Package (SARP), for predicting how useful the user feels to the package, while simultaneously reflecting how much the user is affected by each review. Specifically, SARP jointly considers the relationships of every user, purpose, and review source to learn better latent representations. A self-attention module is further used for integrating source representations and the review ratings, following a multi-layer perceptron (MLP) for the prediction tasks. Experimental results on the self-constructed dataset and public dataset demonstrate that the proposed model outperforms the state-of-the-art approaches.**

***Index Terms*—multi-source review, recommendation system, deep learning**

I. Introduction

With the rapid growth of Internet services, people are overwhelmed with exploding information, and thus recommendation systems become more and more important for users to access the information relevant to their interests. An online service platform that collects information from multiple sources is desirable since the diversified sources satisfying users' needs. For example, Figure 1 illustrates an online booking website, trivago[1], which searches for many hotel booking websites and integrates multi-source reviews about the hotel that the user may be interested in.

In this paper, we aim to find the set of reviews (review package) that is useful to the user. Traditional recommendation systems, such as Factorization Machine (FM) [1]–[4] and auto-encoder [5]–[7], are mostly designed for predicting a single item, and thus fail to recommend a package since the relationship between items is not modeled. For example, when recommending a triathlon-related package, traditional recommendation systems may recommend three similar bicycles as a package if they are the top-3 items. Another line of studies focuses on the bundle recommendation task [8]–[10], which considers the interaction modeling between a user and a set of items. However, the bundle recommendation is different from the multi-source review package recommendation since the interplay between the reviews of multiple sources is not properly modeled. For instance, two reviews from the same sources may conflict with each other, which also affects the usefulness. In contrast, items in a bundle are usually complementary or similar to each other.

[1]https://www.trivago.com/

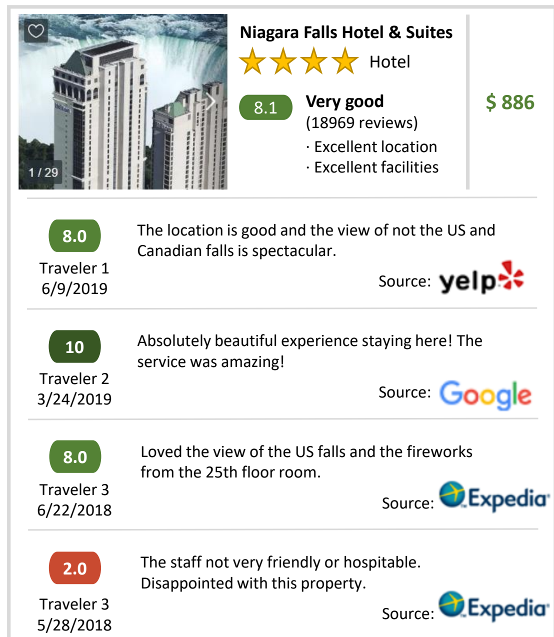


Fig. 1: An illustrative example of the Trivago hotel review.

In fact, predicting the usefulness of a multi-source review package for different users is challenging. First, users have different preferences for consistent and conflicting reviews. For example, some users prefer conflicting reviews since they are more credible and present the viewpoints in different perspectives [11], but others may think that conflicting reviews are confusing since it is easier to make decisions with coherent scores [12]. Second, different users have different preferences on review sources, e.g., some people trust the reviews from google while others only trust the reviews of close friends on Facebook. Third, users' preferences for each source may change under different scenarios [13]. For instance, some of the users may prefer opinions from Local Guide[2] when they go abroad, while others may prefer reviews from Google when they are going to have dinner with family.

To tackle these challenges, we propose Self-Attentive Recommendation for multi-source review Package (SARP) for predicting how useful the user feels about the review package, while simultaneously reflecting how much the user is influenced by each review. Specifically, given the purpose and historical preference of each user on the review package, we propose a tensor factorization-based method [14] to generate the pre-trained embeddings for latent representations of every user, purpose, and source. Afterward, to integrate reviews from multiple sources, we propose to use the self-attention method for learning the different importance of every review in the package and enhance the item embeddings in the package by the attention score.

Moreover, to jointly model the information of user and package, we perform another self-attention computation for package features and user features, while adding a mask to the package features since we focus on the interaction between user and package. SARP further generates the dynamic and static features by using average pooling for enhanced package and user features, respectively. After the concatenation of dynamic and static features, an MLP is used to predict the overall usefulness of the package. On the other hand, SARP also provides the influenceability prediction for each review in the package, i.e., the prediction can tell users which review would be important to them. Therefore, another MLP is used to model the influenceability indicating how much the user is influenced by each review.

Fig. 2 illustrates an example of the prediction task. The relation between usefulness and influenceability depend on the user. Given a package, one may rate the overall usefulness of the review package with 5 stars even though only one of the reviews affects him, while others may rate the overall usefulness of the review package with only 2 stars since not all the reviews affect them. Equipped with the usefulness and influenceability prediction, SARP adopts a multi-task learning strategy to learn a better representation of these two tasks. Experimental results show that SARP outperforms the state-of-the-art method in terms of both tasks.

The contributions are summarized as follows.

[2]https://www.localguidesconnect.com/

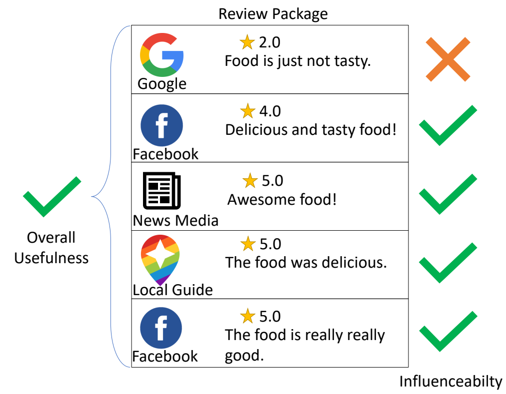


Fig. 2: An illustrative example of MSR dataset and our tasks. A review package in MSR dataset consists of 5 reviews, our task is to predict how much the user is influenced by each review and the overall usefulness about this package.

- We formulate a novel multi-source review recommendation problem and propose a new framework SARP based on a new attention mechanisms to integrate multi-source review data and the user's purpose simultaneously.
- The proposed SARP is able to predict how useful the user feels to the whole package and reflecting how much the user is affected by each review at the same time, which can be adopted to better understand the user behaviors.
- We collect a real-world dataset called **M**ulti-**S**ource **R**eview (**MSR**) dataset[3]. Comprehensive experiments on MSR dataset manifest that SARP outperforms the state-of-the-art methods in both usefulness prediction and the influenceability prediction. Moreover, we evaluate SARP on a public Trivago dataset that consists of user search logs. Experimental results manifest that our model can generalize to the real-world application.

II. Related Work

A. *Context-Aware Recommendation System*

Different from the traditional recommendation systems that consider the user and item information only, context-aware systems that take contextual information into account can deal with more complicated situations, e.g., recommending a tour package based on the travel season [15], personalized content on a website or a restaurant [16]. The contextual information can be known explicitly or implicitly depending on the application scenarios. For example, [17] considers time influence on point-of-interest recommendation, where time information can be explicitly gathered from user's check-in records. However, the user's purchase purpose may not be easily known and need to be modeled implicitly from the user's historical interaction to find his intention. For instance, [18] leverages memory mechanism to store the user's historical records and makes the sequential recommendation.

[3]https://reurl.cc/o9b64q

A recent line of literature exploits Factorization Machine (FM) [1] for context-aware recommendation systems, which models the interaction of two features by inner products to learn the relation between the user, item, and the contextual information [2]–[4], [19], [20]. For instance, FiBiNET [20] dynamically learns the importance of features via a novel Squeeze-Excitation network mechanism and learns the feature interactions via a bi-linear function.

B. Bundle Recommendation

Another related topic focuses on bundle recommendation task [8]–[10], which considers the interaction modeling between a user and a set of items, which can be viewed as a simplified solution of our problem since recommending a bundle of items is similar to modeling contextual information of items. The bundle recommendation task is similar to context-aware tasks since both of them need to deal with feature aggregation problems, but it is slightly different from context-aware since most of the bundle data uses implicit feedback, where only positive feedback (e.g., click, view) is available. Another difference is that number of items in a bundle may not equal, however, what kind of contextual information to be used in context-aware systems is predefined. For example, [21] recommends a list of products to users by optimizing a list's click probability. [10] designs a factorized attention network to aggregate the item embedding in a bundle to obtain the bundle's representation.

C. Multi-Source Recommendation System

A recent line of study focuses on multi-source prediction [22]–[25]. [22] argues that jointly modeling multiple feedback types reveals the underlying spectrum of users' preferences in different dimensions and thus lead to better recommendation performance. Moreover, [23] proposes a model for learning informative representations of users and news by exploiting different kinds of news information. [24] considers the case of user viewing behaviors in multiple web video sites. Nevertheless, none of the above research tackles the challenge of modeling the interplay between different sources. Moreover, our research considers the scenario of conflicting opinions from the same source. To the best of our knowledge, this is the first work modeling these complicated interactions.

III. Proposed Model

A. Problem Formulation

Given an user X_u, his/her purpose or historical behaviors representation X_c, and a package X_p containing a set of review $R_1, ...R_N$, where each review information R_i consists of a source and a rating, the task intends to predict 1) the overall usefulness y_{useful} and 2) individual influenceability $y_{\text{affect};i}$ on how much he/she is affected by the i-th review in this package X_p. y_{useful} and y_{affect} are the users' explicit feedback to a bundle of reviews.

B. General Framework

To precisely predict the usefulness and influenceability, there are three challenges required to be addressed. 1) *Complicated item interplay.* Since each review has a different rating and affects the usefulness of other reviews, simply using a weighted sum for aggregating the items neglects the interplay. How to jointly consider the reviews' information in a bundle is challenging. 2) *Purpose-variant embedding.* Users' preferences for each source may change for different purposes. As such, the user and item embeddings change with different purposes. Deriving the embeddings that model the effect with different situations is difficult and necessary. 3) *Interference of different tasks.* Usefulness and influenceability should facilitate each other. However, without a correct training strategy, usefulness and influenceability can interfere with each other and deteriorate the results.

Keeping the challenges in mind, in this paper, we propose Self-Attentive Recommendation for multi-source review Package (SARP), including three key modules: Tensor factorized embedding layer, Self-attention module, and Usefulness and influenceability prediction module. Figure 3 illustrates the overview model architecture of SARP. Specifically, tensor factorized embedding layer aims to address the cold-start problem, i.e., explicit feedback is available for a few bundles, by factorizing the latent features from the user-source-purpose tensor. To address the first issue, self-attention module for package information aims to jointly learn the interplay among all the items in this package. For the second issue, we leverage another self-attention module to jointly consider the package features and the user features, and model the users' preferences on the reviews in this package. Finally, we leverage the training strategy of multi-task learning to better estimate the usefulness and influenceability for addressing the last challenge.

C. Self-attention for Package Information

To leverage package information in the neural network approach, existing methods usually aggregate all the items into a single vector by either concatenation [2], [19], pooling [26], or attention methods [10]. However, they are not suitable for creating a bundle representation since that concatenation methods may not handle bundles of different sizes, and pooling methods treat all the items in the same way, which cannot learn the difference for all the items. Although some of the methods use the attention mechanism to learn different importance for each item in the bundle and perform weighted sum, the mutual information in the bundle is still not properly modeled.

To extract important information from the reviews and learn better package representation, we leverage the self-attention module [27], [28] for adaptively enhancing the item features. Unlike the vanilla attention simply learns the importance of each review, the self-attention module takes the mutual dependencies between reviews into consideration.

Before performing self-attention on reviews in the package, we integrate the information from each review for better representation. For each review, we have its source and rating information. To obtain a representation for each review, we

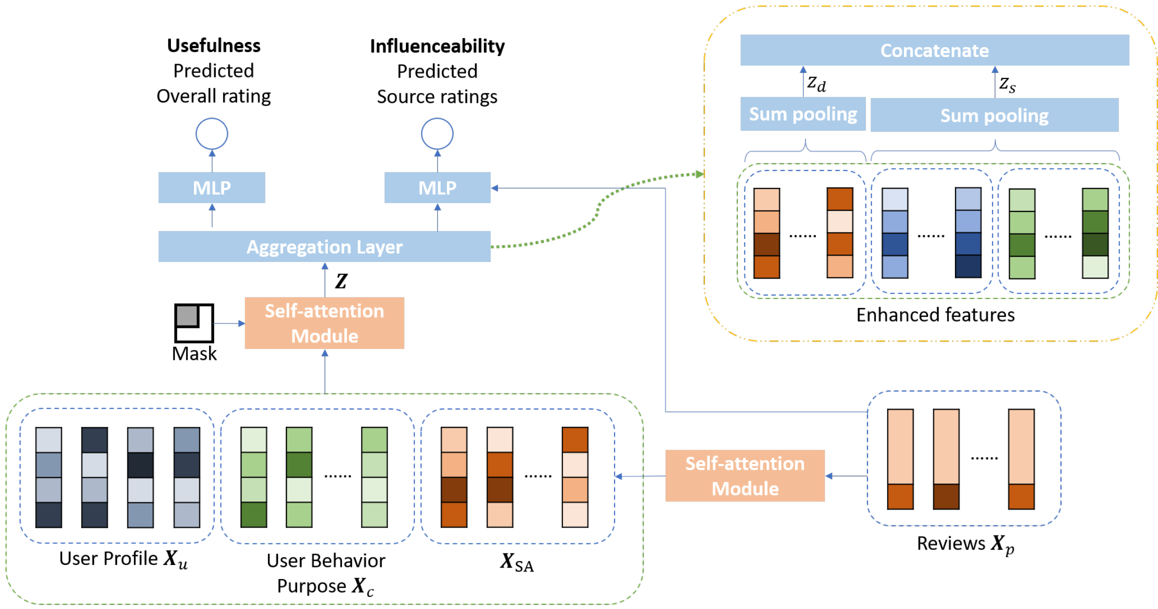


Fig. 3: The overview architecture of our proposed model.

convert the source $\mathbf{s}_i$, which is a one-hot vector, into a dense vector by multiplying with an embedding projection matrix $\mathbf{M}_s$, and simply concatenate the source vector with the rating. Specifically, for the i-th review in a package, we define the review representation R_i as:

$$\mathbf{z}_i^s = \mathbf{s}_i \mathbf{M}_s, \tag{1}$$

$$R_i = [\mathbf{z}_i^s; r_i], \tag{2}$$

where $\mathbf{z}_i^s \in \mathbb{R}^{d \times 1}$ is the source vector, r_i is the rating, and d is the dimension of the representation. Let $\mathbf{X}_p$ denote the package including N reviews:

$$\mathbf{X}_p = \left[R_1^T, R_2^T, \ldots, R_N^T \right]^T. \tag{3}$$

Query, key, and value are 3 inputs of the self-attention module, in our context, the 3 inputs are equal to the package information $\mathbf{X}_p$. After performing non-linear transformation for the 3 inputs, we obtain the enhanced package representation $\mathbf{X}_{SA}$. The procedure of self-attention is as follows:

$$\mathbf{Q} = \tanh(\mathbf{X}_p \mathbf{W}_p^Q), \tag{4}$$

$$\mathbf{K} = \tanh(\mathbf{X}_p \mathbf{W}_p^K), \tag{5}$$

$$\mathbf{V} = \tanh(\mathbf{X}_p \mathbf{W}_p^V), \tag{6}$$

$$\mathbf{X}_{\mathrm{SA}} = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}}\right)\mathbf{V}, \tag{7}$$

where $\mathbf{W}_p^Q, \mathbf{W}_p^K, \mathbf{W}_p^V \in \mathbb{R}^{(d+1) \times d}$ are corresponding weight matrices for the query, key, and value.

As we can see in (7), the output of self-attention is the weighted sum of all values. Namely, the i-th row of $\mathbf{X}_{\mathrm{SA}}$ can be written as $w_{i1}v_1 + w_{i2}v_2 + \ldots + w_{iN}v_N$, where w_{ij} depends on the inner product of i-th query and j-th key, which models the interaction between i-th and j-th components. The i-th feature consists of the mutual information between i-th query and all the other items, which can be regarded as an enhanced package representation $\mathbf{X}_p$.

D. Self-attention for User Information

Let $\mathbf{X}_u \in \mathbb{R}^{N_u \times d}$ and $\mathbf{X}_c \in \mathbb{R}^{N_c \times d}$ be user profile and contextual information, where N_u and N_c are the size of user profile and contextual information. Note that contextual information may also be obtained from user behavior or historical interaction since the user's intentions can be learned from historical records.

We further perform another self-attention mechanism to enhance the package representation, user profile, and contextual information by concatenating these features as

$$\mathbf{X} = \left[\mathbf{X}_{\mathrm{SA}}^T, \mathbf{X}_u^T, \mathbf{X}_c^T \right]^T. \tag{8}$$

Note that we slightly modify the self-attention computation by masking the item features from each other to only model the interplay between package features and user information, i.e.,

$$\mathbf{Q} = \tanh(\mathbf{X}\mathbf{W}^Q), \tag{9}$$

$$\mathbf{K} = \tanh(\mathbf{X}\mathbf{W}^K), \tag{10}$$

$$\mathbf{V} = \tanh(\mathbf{X}\mathbf{W}^V), \tag{11}$$

$$\mathbf{Z} = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}} + \mathbf{M}\right)\mathbf{V}, \tag{12}$$

where $\mathbf{W}^Q$, $\mathbf{W}^K$, $\mathbf{W}^V \in \mathbb{R}^{d \times d}$ are corresponding weight matrices for query, key, and value, and the mask $\mathbf{M} \in \{-\infty, 0\}^{N \times N}$ is a constant matrix that is defined as follows.

$$M_{ij} = \begin{cases} 0, & \text{if } i, j < N_p \\ -\infty, & \text{otherwise} \end{cases} \tag{13}$$

where N_p is the size of the package, and $N = N_p + N_u + N_c$. In other words, the interplay inside the package is neglected by adding the mask $\mathbf{M}$.

E. Usefulness and Influenceability Prediction Module

After computing the enhanced feature $\mathbf{Z}$, we perform average pooling for the first N_p columns as dynamic feature $\mathbf{z}_d$ and the rest of $N_u + N_c$ columns as static feature $\mathbf{z}_s$:

Finally, to predict the overall usefulness of the package, we construct the overall preference representation $\mathbf{z} = [\mathbf{z}_d; \mathbf{z}_s]$ by concatenate the dynamic feature $\mathbf{z}_d$ and static feature $\mathbf{z}_s$ The usefulness of the package is formulated as:

$$\hat{y}_{\text{useful}} = \text{MLP}(\mathbf{z}), \tag{14}$$

It is worth noting that SARP also predicting how much the user is affected by each review. For the i-th review in the package, we consider the review representation R_i and the overall preference representation z, and the predicted result is derived as follows.

$$\hat{y}_{\text{affect};i} = \text{MLP}([\mathbf{z}; R_i]), \tag{15}$$

F. Training Strategy

The loss function of SARP contains two parts. To learn how useful that the user feels under the purpose, we minimize the mean-square error (MSE) between the predicted usefulness and the target usefulness:

$$L_{\text{useful}} = \frac{1}{N_t} \sum_{i=1}^{N_t} (y_{\text{useful}}^{(i)} - \hat{y}_{\text{useful}}^{(i)})^2, \tag{16}$$

where N_t is the size of training data. To learn the user's preference toward the reviews in the package, the loss function is defined as:

$$L_{\text{affect}} = \frac{1}{N_t} \sum_{i=1}^{N_t} \sum_{j=1}^{N_p} (y_{\text{affect};j}^{(i)} - \hat{y}_{\text{affect};j}^{(i)})^2. \tag{17}$$

By introducing the regularization term, the overall objective function of SARP is shown as follows:

$$L = cL_{\text{useful}} + (1 - c)L_{\text{affect}} + \lambda ||\Theta||_F^2, \tag{18}$$

where Θ is the set of parameters in our proposed model, and λ is the regularization coefficient. $||\cdot||_F$ denotes the Frobenius norm. c is the weighting factor. We apply Adam optimizer for adjusting the learning rate adaptively.

IV. Dataset Description

We describe the two datasets used in this chapter.

TABLE I: Dataset statistics for MSR.

# purpose	5	# source	4
# user	304	# record	16782
# rating / record	5	conflicting rating	53 %
avg. rating	3.36 / 5	avg. usefulness	4.09 / 5

A. Multi-Source Dataset

Since there is no public dataset for context-aware multi-source review package recommendation, we introduce a new dataset, **M**ulti-**S**ource **R**eview (**MSR**), collected from real user feedback. The dataset is used to analyze users' preference toward a bundle of reviews, which may come from different sources (websites). Review contents and ratings are related to some restaurants and collected from 4 sources and 5 purposes. Illustrative example is provided in Figure 2.

To construct a review package, 5 reviews are randomly selected while 4 among them are from different sources and the left one is repeated from one of the 4 sources. The two reviews from the same source may have consistent or conflicting ratings to model the user's preference toward the different aspects of reviews. In collecting the MSR dataset, 304 participants were invited to rate review packages. We asked each participant's preferred source among four different review sources. Then, participants were given 12 randomly chosen review packages given a dining purpose and were asked how useful they perceived these packages to be and how much they were affected by each of the individual reviews in a review package. Statistics of MSR are summarized in Table I. 5-fold cross validation is adopted for reporting the results.

By observing from this dataset, the overall rating scores are relatively high on the scale 1-5. The average rating score is 3.36 and the average usefulness score is 4.09. The distribution of the rating score is imbalanced, i.e., 26.48% and 25.32% of all reviews got 5 and 4 respectively on the rating score. Therefore, we define the ratings greater or equal to 4 is defined as positive reviews and otherwise as negative ones [29]. In each package, if the ratings from repeating source are both positive or negative, we define this as a consistent package (46.8 %), otherwise a conflicting package (53.2 %). There are 67.1% of users that prefer consistent reviews, 23.7% prefer conflicting ones, and 9.2% are neutral. Nearly a quarter of users preferring conflicting reviews verifies the need for modeling the interplay of different sources.

B. Trivago Dataset

To demonstrate that our model can capture item interaction in the package and solve the real-world problem, we test our model on a public dataset called **Trivago**, which is provided by ACM RecSys Challenge 2019[4]. The data is from a famous online hotel searching website Trivago and contains users' session actions such as searching for a destination, clicking out an item, etc. As users searching by keywords or selecting filter,

[4]https://recsys.trivago.cloud/challenge/dataset/

TABLE II: Dataset statistics for Trivago.

# user	2615
# item	133127
# click-out record	38211
# impressions	902754
avg. package size	23.63
avg. length of historical records	3.67

the website presents up to 25 hotels based on their needs. The task here is to predict which accommodation in the searching results would be clicked by the user. For each click-out record, we consider its historical interaction with items in the same session for user behavior modeling and take all the hotels presenting in the search results as a package.

How SARP predicts the click-out item in the search result is as the way of predicting influenceabilty in MSR dataset. We rank all the items in the package and predict click-out item. To construct the package representation, we concatenate the item embedding with the price while adding position embedding [27] since the presenting order in the package is important for prediction. The price information about items in the package is processed by Mean Normalization.

Note that we do not evaluate the overall usefulness for the search result since we do not have the information about that. Tensor factorization for pre-trained embedding vectors is also omitted since the prior knowledge about users is needed. Details about Trivago dataset are listed as Table II.

Following [30], we filter out users with less than 5 sessions. For each user, the first 80% of click-out records are selected for training and the remaining data are for testing.

C. Comparison between Two Dataset

Although two datasets are related to different tasks, similar information can be captured from data by our proposed model. In hotel recommendation problem, the self-attention module is used to capture the relation such as price between each of the hotels in the impression list. Another self-attention module for user information is to measure the similarity between user behaviors and the hotels. At last, the MLP module predicts the user preference to each of the hotels. The mutual information between items in the package and the relation between package and user are essential components in both tasks, with the help of the two self-attention modules in SARP, we can complete two tasks without modifying the model architecture.

V. Experimental Results

A. Experimental Settings

Baseline Models We compare our proposed model with the following baselines:

AFM adopts attention mechanism to learn different importance of the pair-wise feature interactions.

Wide & Deep concatenates all the contextual features and then feeds them into a deep neural network. It also considers the cross-product of two features as the input of the linear part. The choice of features to perform cross-product is manually defined.

DeepFM enhances the Wide & deep model by efficiently modeling feature interactions with no need of feature engineering about raw features.

xDeepFM further enhances DeepFM by using vector-level feature interaction.

DIN considers the target item and the rich information of user historical behaviors by a local activation unit, which effectively captures user interests.

DIEN improves DIN by using interest extractor layer to capture temporal interests from historical behaviors.

FiBiNET dynamically learns the importance of features via the Squeeze-Excitation network mechanism and extracts the feature interactions via a bilinear function. FiBiNET obtains state-of-the-art performance in context-aware recommendation tasks.

Evaluation Metrics. We evaluate the performance of usefulness and influenceability prediction using SARP by calculating Mean-Squared Error (MSE) of the usefulness and affect reconstructed vectors.

For Trivago dataset, we evaluate the performance by Mean Reciprocal Rank (MRR), which measures the average ranking of the target item in the list of candidate items. The definition of MRR is as follows:

$$\mathrm{MRR} = \frac{1}{N}\sum_{i}^{N}\frac{1}{\mathrm{rank}_i}, \tag{19}$$

where N is the number of the candidate items. Note that we evaluate Trivago dataset based on the users' preferences for items in the package only, and we do not evaluate the overall usefulness of the whole package.

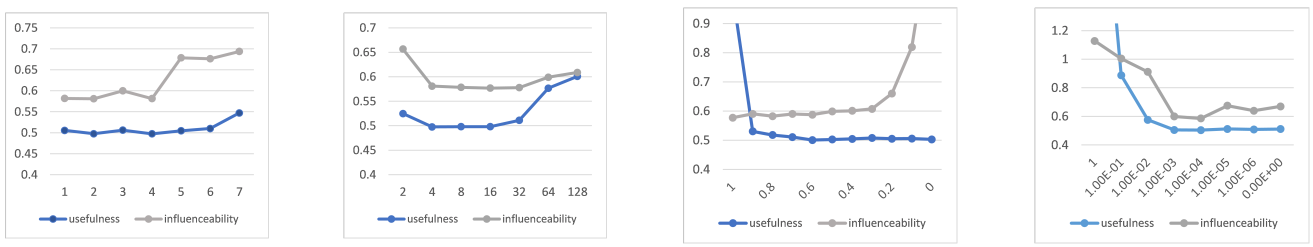
B. Results and Discussions

1) Usefulness and Influenceability: The left side of Table III presents the predicting errors of usefulness and influenceability, which manifests that SARP consistently outperforms the baselines in terms of these two prediction metrics. The MSE error of all baselines based on the modification of *Wide & Deep*, namely *DeepFM*, *xDeepFM*, and *FiBiNET* are high since these models concatenate all features and result in a sparse input vector. They are unable to model the complicated situation in this task. Note that SARP significantly outperforms other baselines in terms of the influenceability error. The reason for the improvement is that SARP deals with individual sources apart from the overall ratings.

2) Capability of Modeling Conflicting Ratings: Moreover, we test the performance on consistent ratings and conflicting ratings divided based on the relationship of repeating reviews from the same source. The right side of Table III shows that the errors of consistent data are mostly lower than those in conflicting data in the same settings, which implies that the conflicting ones are more difficult to model their usefulness and influenceability than the consistent ones. However, SARP still reaches the lowest error on both consistent data and conflicting data, which indicates that the predicted ratings

TABLE III: Performance comparisons of usefulness and influenceability of all methods in terms of MSE. The first two columns on the left indicate the overall performance, following the performance of the consistent part and the conflicting part. "*" indicates the improvement of SARP over the best baseline is significant at the level of 0.01.

	Overall		Consistent Ratings		Conflicting Ratings	
	Usefulness	Influenceability	Usefulness	Influenceability	Usefulness	Influenceability
AFM [3]	0.5599	0.8044	0.5678	0.7878	0.5530	0.8191
Wide & Deep [19]	0.5246	0.7659	0.5142	0.7521	0.5377	0.7780
DeepFM [2]	0.5219	0.7262	0.5104	0.7080	0.5319	0.7422
xDeepFM [4]	0.5300	0.6974	0.5288	0.6912	0.5312	0.7028
FiBiNET [20]	0.5230	0.7254	0.5116	0.7056	0.5330	0.7428
SARP	**0.4976***	**0.5810***	**0.4752***	**0.5585***	**0.5172***	**0.6007***



(a) number of layers (b) embedding dimension (c) task importance factor (d) regularization Coefficient

Fig. 4: Impact of the network hyper-parameters on usefulness and influenceability performance.

TABLE IV: MRR score of all methods. The middle column indicates the publication information about the works. The best method is marked as **bold**. "*" indicates the improvement of SARP over the best baseline is significant at the level of 0.01.

	Venue & Year	MRR
DeepFM [2]	IJCAI 2017	0.4337
xDeepFM [4]	KDD 2018	0.4332
DIN [31]	KDD 2018	0.4391
DIEN [32]	AAAI 2019	0.4370
SARP	-	**0.4496***

TABLE V: Ablation study of the proposed model. The best method is marked as **bold**.

	Usefulness	Influenceability
(1) w/o self-attention for package information	0.5200	0.6045
(2) w/o self-attention for user information	0.5557	0.6086
(3) w/o self-attention	0.5660	0.6109
(4) w/o usefulness prediction module	-	**0.5774**
(5) w/o influenceability prediction module	0.5030	-
SARP	**0.4976**	0.5810

produced by SARP better capture the interplay between the reviews of different sources.

3) Trivago Dataset: Table IV shows the MRR score for click-out item prediction. SARP achieves the best performance among all the baseline since the baseline models cannot properly model the interference from the other items in the same package. They focus on how to model the complicate interaction about user features and the target items, while neglecting the interplay in the package.

C. Ablation Study

To investigate the effectiveness of each component in SARP, we perform the ablation study on different degraded versions of SARP. Results are reported in Table V.

(1) w/o self-attention for package information: The self-attention module for package information is to capture the mutual interaction between items in the package. Removing this module causes to decrease in the performance since the relation between user features and the mutual information of items in the package is not modeled.
(2) w/o self-attention for user information: The self-attention module here is to capture the interaction between user features and package information. After removing the module, a significant drop can be observed. It demonstrates the effectiveness of SARP on leveraging the user's interest in the package.
(3) w/o self-attention: After removing all the self-attention, the performance is further decreasing, which shows the importance of the proposed two self-attention modules for modeling the interaction within the package and the user's interest.
(4) w/o usefulness prediction module: We remove the usefulness prediction part and predict the influenceability only. The performance in this setting is slightly better than the full model, it may show that the usefulness prediction task does not help influenceability prediction a lot, but SARP can still achieve the compatible performance of influenceability.
(5) w/o influenceability prediction module: We remove the influenceability prediction part and do the usefulness prediction task only. The result is slightly worse than the performance in SARP, which may show that the influenceability prediction task may benefit to usefulness prediction task. Note that SARP still achieves the best result in the single task over the baseline models in Table III.

D. Hyper-parameter Analysis

1) Number of MLP Layers: We control the number of layers in MLP and the result is shown in Figure 4a. The performance on influenceability dramatically decreases when the number of layers is larger than 5. We find that it is hard to optimize the model when the number of layers increases. In this dataset, 1 or 2 layers in MLP modules is a good choice.

2) Embedding Dimension: Results for adjusting the embedding dimension are shown in Figure 4b. The user's interest in the multi-source package may not be properly captured when the embedding dimension is less than 4, which is insufficient to represent the user or the source feature. The performance decreases when the dimension is larger than 32 since the overfitting phenomenon.

3) Task Importance Factor: The hyper-parameter c is to balance the importance of two different tasks. The effect of c between 0 to 1 is shown in Figure 4c. SARP would focus on the usefulness prediction task when choosing c that is close to 0 and will focus on the influenceability prediction task when c is close to 1. We choose c to be 0.6 such that both tasks achieve good performance.

4) Normalization coefficient: Adding normalization term avoids the overfitting phenomenon that can be easily seen in machine learning. We choose 0.0001 for the normalization coefficient in the MSR dataset.

VI. Conclusion

In this paper, we formulate a novel multi-source review recommendation problem and present a new model called SARP to effectively model the interplay between reviews from different sources. SARP can predict how useful the user feels to the package, while simultaneously reflecting how much the user is affected by each review. SARP uses two self-attention modules to jointly model the relationships of every user, purposes, and review sources, and jointly considers their interaction to learn better latent representations, following a multi-layer perceptron (MLP) for the prediction tasks. Experimental results manifest that our model outperforms the state-of-the-art approaches.

Acknowledgements

We are grateful to the National Center for High-performance Computing for computer time and facilities. This work was supported in part by the Ministry of Science and Technology of Taiwan under Grants MOST-108-2218-E-009-050 and MOST-109-2221-E-009-097.

References

[1] S. Rendle, “Factorization machines,” in ICDM, 2010, pp. 995–1000.

[2] H. Guo, R. TANG, Y. Ye, Z. Li, and X. He, “Deepfm: A factorization-machine based neural network for ctr prediction,” in IJCAI, 2017, pp. 1725–1731.

[3] J. Xiao, H. Ye, X. He, H. Zhang, F. Wu, and T.-S. Chua, “Attentional factorization machines: Learning the weight of feature interactions via attention networks,” in IJCAI, 2017, pp. 3119–3125.

[4] J. Lian, X. Zhou, F. Zhang, Z. Chen, X. Xie, and G. Sun, “xdeepfm: Combining explicit and implicit feature interactions for recommender systems,” in SIGKDD, 2018, pp. 1754–1763.

[5] H. Wang, N. Wang, and D.-Y. Yeung, “Collaborative deep learning for recommender systems,” in SIGKDD, 2015, pp. 1235–1244.

[6] S. Sedhain, A. K. Menon, S. Sanner, and L. Xie, “Autorec: Autoencoders meet collaborative filtering,” in WWW, 2015, pp. 111–112.

[7] Y. Wu, C. DuBois, A. X. Zheng, and M. Ester, “Collaborative denoising auto-encoders for top-n recommender systems,” in WSDM, 2016, pp. 153–162.

[8] T. Zhu, P. Harrington, J. Li, and L. Tang, “Bundle recommendation in ecommerce,” in SIGIR, 2014, pp. 657–666.

[9] A. Pathak, K. Gupta, and J. McAuley, “Generating and personalizing bundle recommendations on steam,” in SIGIR, 2017, pp. 1073–1076.

[10] L. Chen, Y. Liu, X. He, L. Gao, and Z. Zheng, “Matching user with item set: Collaborative bundle recommendation with deep attention network.” in IJCAI, 2019, pp. 2095–2101.

[11] S.-J. Doh and J.-S. Hwang, “How consumers evaluate ewom (electronic word-of-mouth) messages,” CyberPsychology & Behavior, vol. 12, no. 2, pp. 193–197, 2009.

[12] M. Y. Cheung, C. Luo, C. L. Sia, and H. Chen, “Credibility of electronic word-of-mouth: Informational and normative determinants of on-line consumer recommendations,” International journal of electronic commerce, vol. 13, no. 4, pp. 9–38, 2009.

[13] M. H. Birnbaum and S. E. Stegner, “Source credibility in social judgment: Bias, expertise, and the judge's point of view.” Journal of Personality and Social Psychology, vol. 37, no. 1, p. 48, 1979.

[14] J. D. Carroll and J.-J. Chang, “Analysis of individual differences in multidimensional scaling via an n-way generalization of “eckart-young”decomposition,” Psychometrika, vol. 35, no. 3, pp. 283–319, 1970.

[15] Q. Liu, E. Chen, H. Xiong, Y. Ge, Z. Li, and X. Wu, “A cocktail approach for travel package recommendation,” TKDE, vol. 26, no. 2, pp. 278–293, 2012.

[16] L. Yao, Q. Z. Sheng, Y. Qin, X. Wang, A. Shemshadi, and Q. He, “Context-aware point-of-interest recommendation using tensor factorization with social regularization,” in SIGIR, 2015, pp. 1007–1010.

[17] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. M. Thalmann, “Time-aware point-of-interest recommendation,” in SIGIR, 2013, pp. 363–372.

[18] X. Chen, H. Xu, Y. Zhang, J. Tang, Y. Cao, Z. Qin, and H. Zha, “Sequential recommendation with user memory networks,” in WSDM, 2018, pp. 108–116.

[19] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir et al., “Wide & deep learning for recommender systems,” in Proceedings of the 1st workshop on deep learning for recommender systems, 2016, pp. 7–10.

[20] T. Huang, Z. Zhang, and J. Zhang, “Fibinet: combining feature importance and bilinear feature interaction for click-through rate prediction,” in RecSys, 2019, pp. 169–177.

[21] O. Sar Shalom, N. Koenigstein, U. Paquet, and H. P. Vanchinathan, “Beyond collaborative filtering: The list recommendation problem,” in WWW, 2016, pp. 63–72.

[22] X. Zhou, D. Liu, J. Lian, and X. Xie, “Collaborative metric learning with memory network for multi-relational recommender systems,” arXiv preprint arXiv:1906.09882, 2019.

[23] C. Wu, F. Wu, M. An, J. Huang, Y. Huang, and X. Xie, “Neural news recommendation with attentive multi-view learning,” in IJCAI, 2019, pp. 3863–3869.

[24] H. Yan, X. Chen, C. Gao, Y. Li, and D. Jin, “Deepapf: Deep attentive probabilistic factorization for multi-site video recommendation,” in IJCAI, 2019, pp. 1459–1465.

[25] Q. Zhang, J. Lu and G. Zhang, “Cross-Domain Recommendation with Multiple Sources,” in IJCNN, 2020.

[26] X. He and T.-S. Chua, “Neural factorization machines for sparse predictive analytics,” in SIGIR, 2017, pp. 355–364.

[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in NIPS, 2017, pp. 5998–6008.

[28] T. Chen, H. Yin, Q. V. H. Nguyen, W.-C. Peng, X. Li, and X. Zhou, “Sequence-aware factorization machines for temporal predictive analytics,” in ICDE, 2020, pp. 1405–1416.

[29] S. Park and J. L. Nicolau, “Asymmetric effects of online consumer reviews,” Annals of Tourism Research, vol. 50, pp. 67–83, 2015.

[30] Y. Feng, F. Lv, W. Shen, M. Wang, F. Sun, Y. Zhu, and K. Yang, “Deep session interest network for click-through rate prediction,” in IJCAI, 2019, pp. 2301–2307.

[31] G. Zhou, X. Zhu, C. Song, Y. Fan, H. Zhu, X. Ma, Y. Yan, J. Jin, H. Li, and K. Gai, “Deep interest network for click-through rate prediction,” in SIGKDD, 2018, pp. 1059–1068.

[32] G. Zhou, N. Mou, Y. Fan, Q. Pi, W. Bian, C. Zhou, X. Zhu, and K. Gai, “Deep interest evolution network for click-through rate prediction,” in AAAI, 2019, pp. 5941–5948.