Computer Science and Information Systems 20(3):911–931 https://doi.org/10.2298/CSIS220620002H

Tourism Recommendation based on Word Embedding from Card Transaction Data

Minsung Hong[1], Namho Chung[2]⋆, and Chulmo Koo[2]

[1] Smart Tourism Research Center, Kyung-Hee University
26-6, Kyungheedae-ro, Dongdaemun-gu, Seoul, South Korea
mshong.res@gmail.com
[2] Smart Tourism Education Platform, Kyung-Hee University
26-6, Kyungheedae-ro, Dongdaemun-gu, Seoul, South Korea
{nhchung, helmetgu}@khu.ac.kr

Abstract. In the tourism industry, millions of card transactions generate a massive volume of big data. The card transactions eventually reflect customers' consumption behaviors and patterns. Additionally, recommender systems that incorporate users' personal preferences and consumption is an important subject of smart tourism. However, challenges exist such as handling the absence of rating data and considering spatial factor that significantly affects recommendation performance. This paper applies well-known Doc2Vec techniques to the tourism recommendation. We use them on non-textual features, card transaction dataset, to recommend tourism business services to target user groups who visit a specific location while addressing the challenges above. For the experiments, a card transaction dataset among eight years from Shinhan, which is one of the major card companies in the Republic of Korea, is used. The results demonstrate that the use of vector space representations trained by the Doc2Vec techniques considering spatial information is promising for tourism recommendations.

Keywords: recommender system, word embedding, neural networks, smart tourism

1. Introduction

The information available on the Internet is tremendously and rapidly growing in the big data era [15]. Although it is helpful to people in general, users need a lot of energy and time to find useful information. Therefore, recommender systems have been extensively studied and developed in various domains to provide personalized information such as items, content, and services [3]. In the tourism domain, such systems automatically track tourists' preferences from their explicit or implicit feedback and match the features of tourism items with their needs [6,12]. However, the massive amount of the data available is mainly implicit, such as card payment, sensor, and mobile data, for tourism recommendations [5,17]. Accordingly, it is essential to analyze and apply implicit feedback to tourism recommendations [18]. There is also a data sparsity problem that affects negatively recommendation performance, since it is impossible that tourists generally utilize most tourism items. In addition to these, it is important to properly reflect a location factor (spatial information) in tourism recommender systems [11,16].

⋆ Corresponding author

As stated by [2], a credit or debit card is one of the easiest payment methods in the tourism industry, as confirmed by the increasing number of operators that have adopted card payments. Therefore, many studies [29,11,10] have recently used card transaction data to recommend items for tourists. However, despite using various security techniques, using raw card payment data in the previous study might not be realistic in terms of GDPR (General Data Protection Regulation). That is, the card transaction logs contain a lot of identifiable personal data and make identifying a specific user possible [28].

In this paper, we propose novel recommendation methods based on card transaction data to recommend tourism services to user groups visiting specific tourist locations. The data was statistically processed to protect personal information by a data provider. To avoid the absence of rating scores in the dataset, we model the card transaction data to transform users and items into vector representations using neural network-based word embedding techniques (i.e., Doc2Vec). The vector representations are then used in the content, collaborative filtering, or hybrid-based recommendation algorithm to provide appropriate tourism business services to a user group when they visit a specific destination. Experimental results with around twenty-million statistical card transaction data occurred in Jeju island, one of the most famous tourist attractions in the Republic of Korea, for eight years, demonstrate that the proposed recommendation methods superior to other baseline methods. In particular, several experiments show the positive influences of spatial information on recommendation performance by comparing it with other baseline methods, which are difficult to consider the information in their data modeling. In this regard, our contributions are three-fold as follows:

- We propose competent and serviceable recommendation methods for traveler groups despite the limitation of card transaction data that are statistically processed to protect personal information. Also, they outperform other baseline methods in experiments with real-world huge card transaction data.
- We address the absence of rating scores by introducing Doc2Vec techniques without a specific method. Compared with other baseline approaches based on the RFM method of converting transaction data to rating scores, the proposed methods have better performance even on the evaluation methodology that could be favorable to the approaches.
- In our methods, it is competent to model the preferences of user groups and spatial information simultaneously. Also, it positively influences on recommendation performance, as demonstrated by comparing the methods with recommendation approaches based on Word2Vec techniques.

It is worth mentioning that the proposed methods can be directly applied to raw card transaction data for recommending items to individuals.

2. Related work

2.1. Tourism recommender systems

The tourism industry has grown on a large scale in the past decades, and numerous tourist services have been provided physically and virtually. However, the more significant number of service providers, the more difficult it is to identify and select a suitable tourist item. To reduce the efforts, tourists need to find a tourist item appropriate to their interests. Recommender systems provide items by analyzing tourists' preferences to help them [8,14,9]. In the literature, there are four base recommendation approaches in the tourism industry: content-based, collaborative filtering, domain-specific, and hybrid approaches [22,8]. The content-based approach uses the features of items and users and calculates their similarities to make recommendations. The collaborative filtering approach uses users' past preferences who share similar interests to decide which items to recommend [22]. The domain-specific approach uses various additional information to enhance recommendations such as context, time-sensitive, location, social information, etc [8]. The hybrid method combines these approaches to overcome drawbacks and achieve high recommendation performance [22].

Al-Ghosseinet al. [1] proposed a cross-domain recommender system to address the sparsity problem in hotel recommendations. Their basic idea is that users generally select a destination to visit and then look for a hotel. Therefore, their system considers location-based social networks to learn mobility patterns from hotel check-in data and uses the patterns to recommend hotels. To do this, the authors map items and users from both domains based on a number of observations and learn preferences for regions and hotels. The results are then combined to perform the final recommendation using Bayesian personalized ranking. Hong and Jung [16] developed a multi-criteria recommendation method to recommend restaurants. They consider tourists' nationality as spatial information. A tensor model, which keeps the correlation between its dimensional factors, is exploited to simultaneously model user preferences for multi-criteria, spatial and temporal information. Higher Order Singular Value Decomposition (HOSVD) predicts multiple ratings depending on the spatial and temporal information. The authors analyzed the influences of multiple ratings as well as spatial and temporal information on recommendation performance and revealed the positive efficacy of the factors. A framework, namely, filter-first, tour-second (FFTS), was proposed for addressing complex selection and producing recommendations on a multi-period personalized tour [19]. It considers mandatory Points of Interest (POIs) as well as optional points that tourists optionally visit. The optional points are filtered using an item-based collaborative filtering approach and users' online data. And then, the daily tours are built based on an iterated tabu search algorithm. Pessemier et al. [27] developed the hybrid approach that combines a content-based method handling sparse data, collaborative filtering introducing serendipity, and a knowledge-based approach for pre-filtering, in order to recommend tourist destinations to user groups by aggregating individual recommendations. The authors adopted users' rating profiles, personal interests, and specific demands to provide next destinations. Casillo et al. [7] proposed a knowledge-based approach to search for and recommend tourism services within a knowledge base, which are generated by considering user, context, and service, to individuals. The platform consists of three different points of view, such as the representation of the context, data management & organization, and inferential engines. An oriented and labeled graph model for the representation of Web resources was used.

Unlike the related work mentioned above, except for the last work [7], our approach recommends tourism business services to a user group who visits a specific location. To the best of our knowledge, there were few studies to recommend tourist services in the tourism domain. Similar to the tensor model above-mentioned, we reflect spatial information into modeling user preferences at the same time by using the Doc2Vec technique and consider it in recommendation procedures (i.e., content-based, collaborative filtering, and hybrid approaches).

2.2. Recommendation with word embedding

In the above-mentioned four approaches, there are various methods to make a recommendation, such as a matrix factorization and neural networks. The matrix factorization-based method has been generally applied to real-world recommendation applications due to its high performance, while recently neural networks-based recommender systems have gained considerable interest by overcoming obstacles of conventional models and achieving high recommendation quality [30]. Similar to matrix factorization methods, neural networks-based word embedding techniques from the field of natural language processing field learn low-dimensional vector space representation of input elements [25]. The word embedding learns linguistic regularities and semantics from the sentences and represents the words by vectorized representations [24]. Recently, some of the recommendation methods [24,26,4] used techniques from Word2Vec to represent text-based features, and some of the recommendation algorithms [25,13] applied the techniques to represent items.

Musto et al. [24] empirically compared three word embedding techniques such as Latent Semantic Indexing, Random Indexing, and Word2Vec, on a content-based recommendation. Authors evaluate their methods on MovieLens and DBbook datasets. They map items to textual contents using Wikipedia and use the texts to make recommendations. Also, they aggregate the document representation of items a user liked for generating the user's profile. By exploiting classic similarity measures, available items are ranked according to their descending similarity with respect to the user profile, and top-k items are provided. Baek and Chung [4] developed a multimedia recommendation method using Word2Vec-based social relationship mining. They extract sentiment words from the metadata of multimedia content in TMDb (The Movie Database) and the users' social stream comments about movies. The words are classified through SVM (Support Vector Machine), and Word2Vec techniques are then applied to represent sentiment words into quantifiable vectors. The vector representations of words are used to find a social relationship. They also establish a similarity and trust relationship between users in order to the precise and reliable recommendation of content fitting a user's tendency. Ozsoy [25] also applied the well-known techniques of Word2Vec to recommendation systems. Unlike the above-mentioned work that directly apply the Word2Vec techniques on the textual contents to recommend items, the author uses the techniques to model non-textual contents, the check-ins, for venue recommendation to individual users. In order to model user preferences into a continuous vector representation, the item list in users' visit history is taken sentences into account. Three recommendation algorithms are proposed based on similarities between users and items from the vector representations for users and are evaluated with the Checkin2011 dataset. Esmeli et al. [13] proposed a session-based recommendation using Word2Vec to recommend products. They create product sequences by different session positions and apply the skip-gram method of Word2Vec techniques to calculate similarities between products. Also, they use class imbalance techniques (i.e., synthetic minority over-sampling and under-sampling) to obtain better recommendation performance. They evaluate the proposed method on the RetailRocket dataset.

Like the last two studies, we also consider the lists of items (i.e., tourism business services in card payment transaction data) as sentences to calculate similarities between user groups and the items. In this study, to recommend appropriate services to a user group that visits a specific location, we use the Doc2Vec techniques to reflect spatial information (i.e., tourist destinations) and consider the spatial information in the procedure of making a recommendation list also. The next sections explain how to model card payment transaction data using the techniques and use the trained model in the recommendation process.

3. Modeling card transaction with Doc2Vec

Our objective is to provide top-k tourism services for which a target user group prefers to expenditure, when the group visits a specific destination. In this paper, we use the Doc2Vec techniques namely PV-DM (Distributed Memory version of Paragraph Vector) and PV-DBOW (Distributed Bag of Words version of Paragraph Vector). We used them since they are the primary and initial Doc2Vec techniques. This section briefly introduces the techniques and explains how to model the card transaction data using the methods to achieve the objective.

Doc2Vec techniques were proposed by Mikilov and Le in [20] to create a numeric document representation, regardless of length. It extends the Word2Vec techniques introduced by [23] to go beyond word level to achieve phrase-level or sentence-level representations. It contains two techniques that produce distributed word representations (i.e., word embedding). The representation expresses a word in low dimensional space and carries the semantic and syntactic information of terms and documents [21]. As shown

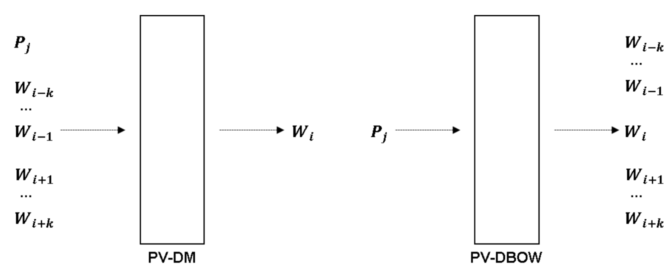


Fig. 1. Doc2Vec techniques

in Fig. 1, the PV-DM technique considers the concatenation of the paragraph vector with the word vectors to predict the next word in a text window, and it is similar to the CBOW (Continuous Bag Of Words) of Word2Vec. While, PV-DBOW predicts the words in a small window, like the skip-gram technique of Word2Vec. The latter one is faster and consumes less memory since there is no need to save the word vectors [20].

We use the Doc2Vec techniques to model card payment transaction data and propose three recommendation methods based on the models trained by the techniques. Therefore, our approach consists of the following two steps. First, the card transaction history is modeled using the Doc2Vec techniques to represent user groups, business services, and locations as numeric vector representations. The outputs are then used to recommend tourism business services to a user group when they visit a specific destination. This section explains the first step. We use the Doc2Vec techniques implemented in the Gensim toolbox [3]. It creates an internal dictionary that holds words and their frequencies, and trains a model using the input data and the dictionary. Its outputs are the vector representations of words and paragraphs. In this paper, the vector representations are considered as the features of user groups, services, and locations.

To model the service usage history of user groups from card transaction data, we generate a list of tourism business services that a user group has used in a specific destination. Destination information is added as a document tag to the list of sentences consisting of words. Therefore, the input data for Doc2Vec techniques indicate the documents, tagged by locations, containing the service usage history of user groups. In Fig. 2, the similarity of input data in the Doc2Vec techniques and recommendation systems is presented conceptually. Fig. 2a presents four sentences in two documents together with the vocabulary

d0, s0: "King walked his cat"
d0, s1: "King took his cat to garden"
d1, s2: "Queen went to garden"
d1, s3: "Queen walked her cat"

Vocabulary:
{king: w0, walked: w1, his: w2, cat: w3, took: w4,
to: w5, garden: w6, queen: w7, went: w8, her: w9}

Conceptual vector:
s0: [1, 1, 1, 1, 0, 0, 0, 0, 0, 0]
s1: [1, 0, 1, 1, 1, 1, 1, 0, 0, 0]
s2: [0, 0, 0, 0, 0, 1, 1, 1, 1, 0]
s3: [0, 1, 0, 1, 0, 0, 0, 1, 0, 1]
d0: [1, 1, 0, 0] d1: [0, 0, 1, 1]

Input data:
s0: [s0, w0, w1, w2, w3, **d0**]
s1: [s1, w0 , w4, w2, w3, w5, w6, **d0**]
s2: [s2, w7, w8, w5, w6, **d1**]
s3: [s3, w7, w1, w9, w3, **d1**]

Output data:
d0: [f0, f1, ..., fn], d1: [f0, f1, ..., fn]
s0: [f0, f1, ..., fn], s1: [f0, f1, ..., fn]
s2: [f0, f1, ..., fn], s3: [f0, f1, ..., fn]
w0: [f0, f1, ..., fn], w1: [f0, f1, ..., fn]
w2: [f0, f1, ..., fn], ... w9: [f0, f1, ..., fn]

(a) Document-Sentence-word data

loc0, user0: ser0, ser1, ser2, ser3, ser4, ...
loc0, user1: ser1, ser5, ser6, ser9, ...
loc1, user2: ser4, ser7, ser8, ...
loc1, user3: ser2, ser3, ser5, ser7, ser9, ...

Vocabulary:
{ser0, ser1, ser2, ser3, ser4,
ser5, ser6, ser7, ser8, ser9, ...}

Conceptual Vector:
user0: [1, 1, 1, 1, 1, 0, 0, 0, 0, 0]
user1: [0, 1, 0, 0, 0, 1, 1, 0, 0, 1]
user2: [0, 0, 0, 0, 1, 0, 0, 1, 1, 0]
user3: [0, 0, 1, 1, 0, 1, 0, 1, 0, 1]
loc0: [1, 1, 0, 0] loc1: [0, 0, 1, 1]

Input data:
user0: [u0, s0, s1, s2, s3, s4, ..., **l0**]
user1: [u1, s1, s5, s6,s9, ..., **l0**]
user2: [u2, s4, s7, s8, ..., **l1**]
user3: [u3, s2, s3, s5, s7, s9, ..., **l1**]

Output data:
l0: [f0, f1, ..., fn], l1: [f0, f1, ..., fn]
u0: [f0, f1, ..., fn], u1: [f0, f1, ..., fn]
u2: [f0, f1, ..., fn], u3: [f0, f1, ..., fn]
s0: [f0, f1, ..., fn], s1: [f0, f1, ..., fn]
s2: [f0, f1, ..., fn], ...

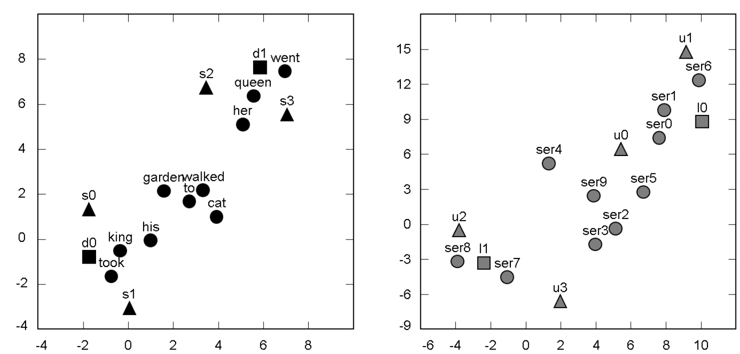
(b) Location-user-service data

Fig. 2. Data examples for Doc2Vec and recommendation

list (dictionary). Similarly, four user groups and the lists of business services, which the groups paid at two specific destinations, are presented in Fig. 2b. Similar to the example in the left figure, it is possible to create a list of services for a user group that visited a specific location. Consequently, both examples are represented as vectors started with sentence and user group identifications followed by word or service ones. The service order is equal to the usage sequence of a user group. Each vector as an input one is added a corresponding document or location tag. Inspiring from [25], the lists of items (i.e.,

[3] `https://radimrehurek.com/gensim/models/doc2vec.html`

services), user groups, and locations are used together as the input data of Doc2Vec techniques. As a result, their vectors are obtained separately and are able to be utilized to decide on which element (i.e., a service, user, or destination) is contextually closer to which elements. Accordingly, documents are abstractly separated into sentence and user group levels. Consequently, input data for the Doc2Vec techniques are constructed by sentences and user groups. Finally, elements' representation vectors trained by Doc2Vec techniques contain n real numbers as shown at the bottom of Fig. 2. Fig. 3 presents the output of PV-DM technique on the data example given in the above figure. To plot, the output vector representations with n dimension (i.e., the feature parameter above-mentioned) were converted into two dimensions using principal component analysis. In Fig. 3a, the output



(a) Document-Sentence-word vector **(b)** Location-user-service vector

Fig. 3. Vector representation of data examples for Doc2Vec and recommendation

for document-sentence-word data is shown. According to this figure, the relations among documents, sentences, and words are captured. Note that we add the sentence IDs at the first position of input data to obtain the relations of the sentences with other elements, as shown in Fig. 2. For instance, while the "king" and "his" are closer to the document "d0" and sentences "s0" and "s1", the word "queen", "her", and "went" are closer (i.e., more related) to the document "d1" and sentences "s2" and "s3". Remind that these words are seen only in these each document and corresponding sentences. The "walked", "to", and "garden" are closer (more related) to each other and located in the middle of both documents since they appear in the documents. These results indicate that the PV-DM technique is able to capture the relations between these documents, sentences, and words. In Fig. 3b, the output for location-user-service data is presented. From the figure, relations among the elements are able to be observed. For example, services "ser0" and "ser6" are represented closer to user groups "u0" and "u1" respectively and located nearby location "l0". The user groups utilize the "ser0" and "ser6" at the location "l0". Other examples are the relations among services. The "ser2" and "ser3" are closer and are always used together in the input data, even in different locations.

4. Recommendation using vector representation

The Doc2Vec techniques provide elements in vector space where similar elements are located closer to each other, as presented in the previous section. We apply the results to three recommendation algorithms.

K-nearest business-based method (KNB) belongs to the content-based recommendation approach. In the traditional approach, item features are used to recommend other items similar to what a user likes. Therefore, we calculate the similarity between a target user group, a specific destination, and services using the vector representations resulting from Doc2Vec. Cosine similarity allowed by the Doc2Vec of the Gensim library is used. As a result, the most similar k services to the target user group and the destination are recommended to the group. Algorithm 1 presents these processes. For instance, given the

Algorithm 1: K-nearest business-based recommendation

Data: Vector representations for services S, a user group U_i, and a destination L_j
Result: List I of top-K services

Set K for # of recommendation
Initialize an empty lists I with the length K and Is with the length S
Calculate a simple mean T_{ij} of the projection weight vectors of the U_i and L_j
for k = *1 to the length of* S **do**
    Calculate cosine similarity cos_{ijk} between S_k and T_{ij} and append it into the Is
end
Sort the list Is in descending order with keeping indexes
Put the K services corresponding first K elements' indexes from the Is into the I
Return I

vectors presented in Fig. 3b, assume that we want to offer two services that are not used by the user group "u0". The most similar services to the user group are "ser5" and "ser9" except for the services already used by the group, so these services are recommended to the target user group. Note that the services used by a user group in past are provided in real, since a user group often uses a business service again.

N-nearest users method (NNU) applies the traditional user-based collaborative filtering approach to the vector representations modeled in the previous step. In the traditional approach, first, the most similar users (neighbors) to a target user are found, and the items preferred by the neighbors are provided to the target user. Similar to the approach, we first decide on N neighbors using the similarities among vector representations of a target user group and a specific location (i.e., similar neighbors visited the location). The services previously used/preferred by the neighbors are then collected. Finally, top-k services are selected to recommend by summing up the neighbor votes, as shown in Algorithm 2. For example, using the example presented in Fig. 2b, assume that we want to recommend two services to the user group "u3" visited at the location "l0", by using two neighbor groups. According to vector representations in Fig. 3b, the most similar user groups, "u0" and "u1", are selected as the neighbors. The service "ser1" previously visited by both of the groups and another service chosen randomly among the services utilized by "u0" or "u1" (i.e., “ser0”, “ser2”, “ser3”, “ser4”, “ser5”, “ser6”, and “ser9”) are recommended to the target user group “u3” located at the “l0”.

Algorithm 2: N-nearest users-based recommendation

Data: Vector representations of user groups U and a destination L_j, use history $\mathcal{H}$
Result: List I of top-n services

/* Define variables */
Set N for # of neighbor groups and K for # of recommendation
Initialize an empty lists NU with the length N and NUs with the length U
Initialize an empty list I with the length K
Calculate a simple mean vector T_{ij} of the target user group U_i and L_j
/* Calculate cosine similarity for neighbor groups */
for k = *1 to the length of* U **do**
  if $k \neq i$ **then**
    Calculate cosine similarity cos_{ijk} between U_k and T_{ij}
    Append the cos_{ijk} into the list NUs
  end
end
/* Make top-N neighbor group list */
Sort the list NUs in descending order with keeping indexes
Put the first N elements' indexes from the NUs into the NU
/* Collect services used by the neighbor groups */
for k = *1 to* N **do**
  Append services used by NU_k from $\mathcal{H}$ into service pool list $Itemp$
end
/* Get top-K services by summing up the votes of the neighbor groups */
Sort the list $Itemp$ by service frequency in descending order and remove duplicates
Put the first K services from the $Itemp$ into the I
Return I

N-nearest users and k-nearest business method (NKB) is a hybrid method of the previous two methods. In NKB, N neighbor groups are first found by using the vector representations of a target user group and a specific location. And then, we search for the top-k services that are the most similar to the combination of the user groups, which consist of the target group and the neighbor groups, and the location. The collected top-k services are recommended to the target user group visited at the location, as shown in Algorithm 3. For example, assume that we want to recommend three services to the user group “u0” visited at the location “l0” using a single neighbor. The user group “u1” would be selected as the neighbor based on the vector similarity. The three most similar services to the user groups “u0” and “u1” as well as location “l0” are “ser0”, “ser1”, and “ser6”. These three services are provided to the target user group “u0” visited the location “l0” by the NKB method.

Our methods can handle the cold-start problem for new user groups that have never used any services in our system since the Doc2Vec techniques also result in vector representations of locations, as shown in Fig. 3b. For instance, when a new user group requests to recommend services in a specific location, our methods can find services with the most similar vector representations to the location's vector representation or search for neighbor groups based on their vector representations.

Algorithm 3: N-nearest and k-nearest business-based recommendation

Data: Vector representations of user groups U and a destination L_j, use history $\mathcal{H}$
Result: List I of top-n services

/* Define variables */
Set N for # of neighbor groups
Set K for # of recommendation
Initialize an empty lists NU and NUs with the length N
Initialize an empty lists I with the length K and Is
Calculate a simple mean vector T_{ij} of the target user group U_i and L_j
/* Calculate cosine similarity for neighbor groups */
for k = *1 to the length of* U **do**
  if $k \neq i$ **then**
    Calculate cosine similarity cos_{ijk} between U_k and T_{ij}
    Append the cos_{ijk} into the list NUs
  end
end
/* Make top-N+1 neighbor group list including the target user group */
Sort the list NUs in descending order with keeping indexes
Put the vectors corresponding first N elements' indexes from the NUs into the NU
Put the vectors of user group U_i into the neighbor list NU
Calculate a simple mean vector T_j of the user groups in NU and location L_j
/* Calculate cosine similarity for services */
for m = *1 to the length of* S **do**
  Calculate cosine similarity cos_{jm} between S_m and T_j
  Append the cos_{jm} into the list Is
end
/* Get top-K services based on vector similarity */
Sort the list Is in descending order with keeping indexes
Put the K services corresponding first K elements' indexes from the Is into the I
Return I

5. Experimental design

5.1. Dataset

To evaluate the proposed recommendation methods, we use a transaction dataset of credit and debit cards from Shinhan card, one of the major card companies in the Republic of Korea. The dataset consists of transaction logs that happened on Jeju island, one of the most famous tourist attractions in the country. It contains 19,648,116 card transactions. As mentioned above, all identifiable personal data were statistically processed to make them anonymous by considering GDPR. Accordingly, there are 1,260 user groups categorized by gender, age groups, habitation cities, nationalities, and time periods of card usage, as listed in Table 1. Tourism services categorized by KSIC (Korea Standard Industry Code[4]), based on the International Standard Industrial Classification (ISIC) adopted by the UN, are taken into account. Thereby, 413 services related to the tourism domain are selected, such as retail, wholesale, accommodation, restaurant, and transport businesses. Also, we have 79 destinations since all card transactions in the dataset happened in Jeju and Seogwipo tour cities. Finally, we use around fifteen million transaction data and split it by an 80-10-10 ratio for training, validation, and test sets, respectively. For the recommendation methods based on the techniques of Word2Vec and Doc2Vec, we obtained 60,410 and 47,847 sentences as training and validation sets.

Table 1. Statistic information of preprocessed dataset

Feature	Number	Feature	Number
# of transaction group	14,673,210	# of user groups	1,260
# of training set	11,738,568	# of tourism business services	413
# of validation set	1,467,321	# of destination	79
# of testing sets	1,467,321		

5.2. Evaluation measure

This section introduces a segmentation technique used in the marketing field, namely RFM, to convert card payment transactions into rating scores that make the comparison of the proposed methods with other baseline approaches feasible. In other words, baseline methods are based on the rating scores to recommend items, unlike the proposed methods. Also, several measures to evaluate them on top-N recommendation are explained.

Inspiring by [29], the RFM method which, is an instrument for analysis in marketing, is used along with k-means clustering technique to determine the ratings. The RFM indicates recency, frequency, and monetary defined as follows:

- **Recency** is calculated by $R = M + (12 \times (Y - Y_b))$.
- **Frequency** presents the number of transactions per user group.
- **Monetary** means the total amount of transaction per user group.

For the recency factor, Y_b and Y indicate the initial year of transactions contained in our dataset and the year of the corresponding transaction of each user group, respectively. We set $Y_b = 2012$ since our dataset contains card payment data occurred from 2012 to 2019. To combine these three features, we use different weights according to their significance level. We set the weights of recency, frequency, and monetary as 1, 2, and 4, by following [29]. As presented in Algorithm 4, we generate ratings as labels for each transaction that is statistically processed to protect personal information. First, we remove the top 1%

[4] KSIC: `http://kssc.kostat.go.kr/ksscNew_web/ekssc/main/main.do`

Algorithm 4: Calculation ratings by RFM method

Data: Transaction data T, feature weights
Result: Labeled transaction data $\hat{T}$

Copy data from T to $\hat{T}$
`/* Set the number of clusters */`
Set $k = 5$
Remove top 1% records for frequency and monetary features
`/* Get labels for each feature */`
for *Each feature (i.e., recency, frequency, and monetary)* **do**
    Run k-means clustering with k to get initial labels
    Reorder the labels based on the clusters' mean values by ascending order
    Add the weighted label into $\hat{T}$
end
`/* Calculate ratings with features' labels */`
Run k-means clustering with k for three feature labels to get final labels
Reorder the final labels based on the clusters' mean values by ascending order
Add ratings into $\hat{T}$
Remove the labels for the three features from $\hat{T}$
Return $\hat{T}$

records as outliers or genuine bulk buyers. For each feature above, we then get labels using the k-means clustering method, implemented in the Sklearn Python library[5], with the cluster number k. We set k as 5 to generate rating scores scaled from 1 to 5 by the above RFM method. The labels resulted by the clustering are reordered by the periods of clusters' mean values with ascending order. To merge the feature labels, we multiply them with corresponding weights. Finally, k-means clustering is conducted with the weighted three features' labels (i.e., multiple feature clustering), and the final labels are reordered to obtain ratings of each transaction.

Since the rating scores are artificially made by the RFM method, we utilize the rank-based evaluation measurements instead of the RMSE and MAE that are directly based on the artificial scores. Among evaluation measures used in this paper, first of all, we introduce MRR (Mean Reciprocal Rank) defined by

$$MRR(U) = \frac{1}{|U|} \sum_{u \in U} \frac{1}{k_u}, \tag{1}$$

where U and k_u indicate a set of users and a rank of the first relevant item for a user u. This measure is simple to compute and easy to interpret. Also, it focuses on only a single item from the list since it puts a high focus on the first relevant element of the list. Although this might not be a good evaluation metric for users who want a list of related items to browse, we consider it since a small number of services are in general used by tourists in the tourism industry, as shown in Table 2. In the table, we can see more 30% user groups used less than five services.

[5] `https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html`

Table 2. Percentages of travelers by the number of used services

# of used services	< 5	< 10	< 20	< 30	< 40	< 50	$50 >=$
Percentage	33.31	19.25	18.78	10.34	6.71	4.20	7.41

To consider multiple relevant items, we use $MAP@k$, which is the mean of $AP@k$, which calculates an average $P@k$ for a user, for all the users. And, $P@k$ measures the relevance of items on k recommended items and is defined as follows:

$$P@k = \frac{|R|}{k}, \tag{2}$$

where R refers to relevant items on top-k item list. It is a simple way to know the fraction of relevant items that are good. However, $MAP@k$ is unable to consider the recommended list as an ordered list, since $P@k$ treats all the errors in the recommended list equally.

Therefore, we use mAP (Mean Average Precision). Unlike the above $AP@k$, $AveP@k$ has the ability to reflect the order of a recommendation list. The mAP is the average of the $AveP@k$ that is defined by

$$AveP@k = \frac{1}{|R|} \sum_{i}^{k} P@i \times rel@i, \tag{3}$$

where R refers to relevant items on top-k item list, and $P@i$ indicates precision at i. The $rel@i$ is a relevant function that returns 1 if the item at rank i is relevant and 0 otherwise. This measure is able to handle the ranks of lists recommended items naturally and shines for binary (relevant/non-relevant) ratings. However, it is still not fit for fined-grained numerical ratings.

In this regard, we also consider NDCG (Normalized Discounted Cumulative Gain), which is able to use the fact that some items are more relevant than others. In other words, highly relevant items should come before medium relevant items, which should come before non-relevant items. This metric is calculated by DCG_k and $IDCG_k$ defined as follows:

$$\begin{aligned} DCG_k &= \sum_{i=1}^{k} \frac{2^{rel_i} - 1}{\log_2(i+1)} \\ IDCG_k &= \sum_{i=1}^{|REL_k|} \frac{2^{rel_i} - 1}{\log_2(i+1)} \\ nDCG_k &= DCG_k / IDCG_k, \end{aligned} \tag{4}$$

where rel_i is the graded relevance of the results at position i (i.e., gain), and REL_k refers to a list of relevant items ordered by their relevance up to top-k. Also, the logarithmic reduction factor is added to penalize the relevance score proportionally to item positions.

5.3. Baseline and variant methods

This section introduces baseline approaches compared with our methods in this study. The baselines consist of two approaches [25,29]. It is difficult to directly apply traditional rating-based methods to the card transaction data since the card payment dataset does not include rating information. Therefore, [29] exploited the RFM method to create the ratings of users' transaction groups and applied the collaborative filtering methods. We compare our methods with the approach using GSVD, SVD, and NMF. The author utilized GSVD and SVD in [29]. We call them $GSVD_{RFM}$, SVD_{RFM}, and NMF_{RFM}. Note that we only use the RFM method described in Algorithm 4 to compare our methods with the baselines, not to make recommendations. Another approach in [25] uses Word2Vec as mentioned in Section 2.2. The author proposed three techniques as content-based, collaborative filtering, and hybrid-based recommendations. These methods are named KNI_{W2V}, NN_{W2V}, and KIU_{W2V}. We use these techniques to show the effectiveness of considering location information on a top-k recommendation. Note that we only considered the skip-gram technique of Word2Vec due to its better performance on our dataset.

Furthermore, we evaluate the proposed methods' variants to reveal the effectiveness of location information in both the Doc2Vec-based data modeling and the top-k recommendations. We proposed three algorithms in Section 4. Also, two Doc2Vec techniques (i.e., PV-DM and PV-DBOW) are used to model data, and we add prefixes $_{DM}$ and $_{DB}$ for the techniques, respectively. Additionally, location information is considered in only data modeling based on Doc2Vec or in the processes of modeling data and making recommendations. These are distinguished by using prefix $_m$ and $_b$. For example, DM-based NNU with location information for both is annotated as NNU_{DM_b}.

We implemented all the above methods using Python and evaluated them in the same experimental environments with the same data sets. First, the validation and test sets were used to determine optimal parameters for each method in a grid search fashion. Using the optimal parameters, we trained the models of all the methods on the training set. Finally, the experimental results on the test set represented in the next section were obtained.

6. Evaluation and discussion

6.1. Comparison of variants

This section evaluates the variants of proposed methods to comprehend the effectiveness of considering location information in modeling and recommendation processes.

Several parameters affect data modeling and result in recommendation performance. These parameters are based on the Gensim toolbox implementation. In this paper, only four parameters are set to a different value from the default in the toolbox. The rest of the parameters are set as the same as presented on the Gensim web page[6]. The details of the parameters and how we tune them are as follows:

- *min_count* ignores the items whose frequency is less than it. Data in recommender systems is very sparse and contains many items observed only a few times in general. To prevent the loss of these items, we set this parameter as one during our experiments.
- *vector_size* represents the dimension of representation vectors, and its default is 100. We empirically set it to different values in the range of [5:50] with 5 increments.

[6] https://radimrehurek.com/gensim/models/doc2vec.html

- *window* assigns the maximum distance between the current and predicted words. It should be large enough to recognize the semantic relationships between words. We test this parameter with different values in the narrow range of [2:20] with 2 increments.
- *epochs* parameter indicates the number of iterations on modeling input data, and its default is 10. In our experiments, it sets to various values in the range of [5:50] with 5 increments.

We conducted a grid search for all their combinations on validation and test sets to find an optimal set of these parameters for each variant. Table 3 lists the performance results and optimal parameter settings of the variants based on the models trained by Doc2Vec techniques. According to Table 3, it can be aware that considering the location information

Table 3. Performance results of variants with optimal parameters

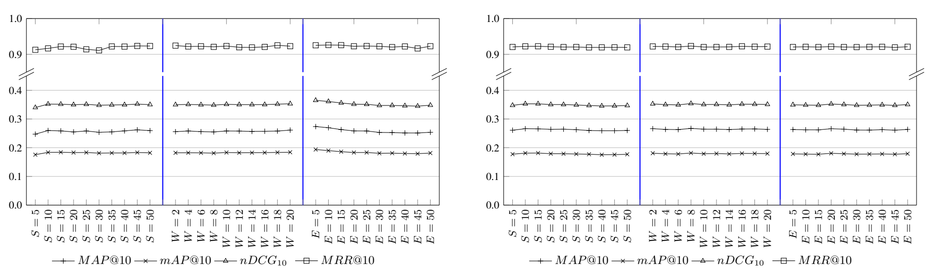
Variants	$MAP@10$	$mAP@10$	$mNDCG_{10}$	$MRR@10$	Optimal setting
KNB_{DM_m}	0.0495	0.0159	0.0495	0.1330	V: 5, W: 10, E: 10
NNU_{DM_m}	0.1868	0.1034	0.2269	0.6071	V: 20, W: 10, E: 10
NKB_{DM_m}	0.0482	0.0160	0.0493	0.1355	V: 5, W: 10, E: 10
KNB_{DB_m}	0.0124	0.0023	0.0095	0.0179	V: 50, W: 2, E: 25
NNU_{DB_m}	0.1853	0.1086	0.2270	0.5902	V: 10, W: 8, E: 15
NKB_{DB_m}	0.0100	0.0016	0.0087	0.0152	V: 50, W: 6, E: 25
KNB_{DM_b}	0.1374	0.0588	0.1408	0.3072	V: 40, W: 8, E: 15
NNU_{DM_b}	0.2843	0.2009	0.3748	0.9287	V: 20, W: 4, E: 5
NKB_{DM_b}	0.1180	0.0504	0.1225	0.2794	V: 40, W: 8, E: 15
KNB_{DB_b}	0.0647	0.0199	0.0639	0.1709	V: 50, W: 10, E: 15
NNU_{DB_b}	0.2673	0.1816	0.3539	0.9231	V: 10, W: 8, E: 25
NKB_{DB_b}	0.0461	0.0132	0.0446	0.1213	V: 5, W: 10, E: 8

[a] V, W, and E indicate parameters *vector_size*, *window*, and *epoch*.

in both modeling data and making recommendations has a lot of improvements from applying only in data modeling.

In terms of Doc2Vec techniques, when we consider location information in only data modeling, data modeling based on PV-DM has more positive influences than that of PV-DBOW. These results are more clear when we compare it with those of the KNI_{W2V}, NN_{W2V}, and KIU_{W2V} in Table 5. The variants considering the spatial information in only modeling recommend services based on vector representations of user groups like the baseline methods. We use Doc2Vec techniques to model users' preferences and spatial information simultaneously, but the models in [25] consider only users' preferences based on the techniques of Word2Vec. The KNB_{DM_m} and NKB_{DM_m} have better performance than the KNI_{W2V} and KIU_{W2V}, while the performance results of the KNB_{DB_m} and NKB_{DB_m} are worse than them (refer to Table 5). In fact, these results are related to the Doc2Vec implementation of Gensim toolbox. The PV-DBOW trains only document vectors with the default setting for *dbow_words*, and it means that the KNB_{DB_m}, NNU_{DB_m}, and NKB_{DB_m} may be unable to appropriately consider the individual history of user groups on the recommendation process. As a result, they show lower performance than the KNB_{DM_m}, NNU_{DM_m}, and NKB_{DM_m}, respectively. Even though we had actually evaluated the PV-DBOW with *dbow_words* $= 1$, which set the technique works in the skip-gram fashion, we couldn't discover remarkable performance differences. Therefore, we presented the performance results based on the pure PV-DBOW technique in this paper. However, the KNB_{DB_b}, NNU_{DB_b}, and NKB_{DB_b} considering the location information in both procedures are superior to the KNI_{W2V}, NN_{W2V}, and KIU_{W2V}. It implies that considering location information in tourism service recommendations is important. Regarding recommendation algorithms, all methods, regardless of Doc2Vec techniques and the consideration of location information, show similar trends of performance results. The NNU methods are superior to the others in all evaluation metrics, and the KNB methods perform better than the $NKBs$. We carefully guess that the reason is caused by the construction process of input data to model card transaction data. Because a user group's identification locates as the first term in the input, the services placed at the beginning have similar vector representations with the user group due to the principle of Doc2Vec techniques. Consequently, KNB, which directly searches for similar services with a target user group, could have a severe bias between services according to their locations in the input data. It also happens to the NKB. Accordingly, we select the NNU_{DM_b} and NNU_{DB_b} to compared with baseline methods in the next section.

To comprehend the effects of three parameters for Doc2Vec techniques on top-10 recommendations, this section evaluates the proposed methods (i.e., NNU_{DM_b} and NNU_{DB_b}) by changing the Doc2Vec parameters in the corresponding ranges mentioned above. While repeating the different ranges of one parameter, the others are fixed to constant values. Figures 4a and 4b show the performance of NNU_{DM_b} and NNU_{DB_b} by parameter value. NNU methods seem to be not affected by the three



(a) Performance result of NNU_{DM_b}

(b) Performance result of NNU_{DB_b}

Fig. 4. Performance results of NNU

parameters unlike the other variants in our preliminary results (omitted due to limited space). To clearly reveal the effects of these parameters on these methods, we analyzed the correlation between the parameters and performance, as listed in Table 4. The correlations indicate that the increase of $vector\_size$ and $window$ parameters positively affects while

Table 4. Correlation analysis between parameters and performance for NNUs

NNU_{DM_b}	*v_size*	*window*	*epoch*	NNU_{DB_b}	*v_size*	*window*	*epoch*
MAP@10	0.535	0.612	-0.903	MAP@10	-0.695	-0.160	-0.229
mAP@10	0.267	0.709	-0.879	mAP@10	-0.700	-0.068	-0.023
MRR@10	0.552	-0.076	-0.752	MRR@10	-0.789	-0.086	-0.134
$mNDCG_{10}$	0.393	0.611	-0.897	$mNDCG_{10}$	-0.712	-0.104	-0.093

the epoch increment has a negative influence on the recommendation performance of the NNU_{DM_b} method. Whereas, in the case of NNU_{DB_b} method, the increments of all the parameters result in worse performance. Indeed, we observed that the proposed methods based on the PV-DM technique are positively affected by the increment of the $vector\_size$ and $window$ parameters, while the proposed methods based on PV-DBOW have mostly negative influences by the increase of all three parameters. However, as shown in the experimental results of previous sections, we need to explore the parameter combinations rigorously to set the optimal one. According to Figure 4 and the results discussed above, the NNU methods have the best performance in general. Consequently, we select the NNU_{DM_b} and NNU_{DB_b} to compared with baseline methods in the next section.

6.2. Comparison with baselines

In this section, we compare the performance results of the two NNU methods based on the pre-trained card transaction data (i.e., the representation vectors of users and items, locations) and the baseline approaches mentioned in Section 5.3. Table 5 presents their performance on various top-k recommendations. The bold and italic font styles indicate the first and second-best performance. According to this table, the proposed methods are superior to the other baselines in most performance measures. Interestingly, the Word2Vec-based approaches (i.e., KNI_{W2V}, NN_{W2V}, and KIU_{W2V}) show worse performance than the other baseline methods. These results might be because of the adopted evaluation methodology. It is based on the RFM, which makes it possible to work matrix

Table 5. Performance results of the proposed and baselines methods

Methods	Top-10				Top-5				Top-2			
	MA	mA	mN	MR	MA	mA	mN	MR	MA	mA	mN	MR
$GSVD_{RFM}$	0.170	0.106	0.214	0.529	0.137	*0.262*	0.093	0.165	0.117	0.190	0.101	0.117
SVD_{RFM}	0.131	0.082	0.176	0.492	0.125	0.239	0.087	0.150	0.097	0.164	0.083	0.106
NMF_{RFM}	0.107	0.059	0.135	0.378	0.072	0.233	0.053	0.094	0.091	0.165	0.085	0.105
KNI_{W2V}	0.029	0.007	0.023	0.042	0.008	0.014	0.003	0.007	0.003	0.004	0.002	0.003
NN_{W2V}	0.124	0.072	0.148	0.355	0.084	0.211	0.061	0.101	0.079	0.121	0.074	0.086
KIU_{W2V}	0.032	0.008	0.028	0.070	0.010	0.038	0.008	0.014	0.017	0.033	0.017	0.020
NNU_{DM_b}	**0.284**	**0.201**	**0.375**	**0.929**	**0.366**	**0.265**	**0.426**	**0.862**	**0.294**	**0.277**	**0.341**	**0.535**
NNU_{DB_b}	*0.267*	*0.182*	*0.354*	*0.923*	*0.274*	0.224	*0.363*	*0.841*	*0.292*	*0.276*	*0.339*	*0.534*

[a] MA, mA, mN, MR refer to the MAP, mAP, $mNDCG$, and MRR, respectively.

factorization-based collaborative filtering approaches on card payment transaction data, and is used to obtain a ground truth set. In other words, the methodology could be favorable to the RFM-based methods (i.e., $GSVD_{RFM}$, SVD_{RFM}, and NMF_{RFM}). Despite this, the proposed methods NNU_{DM_b} and NNU_{DB_b} outperform the other methods. It indicates that our methods can adequately model the card transaction data by considering the spatial information and to make appropriate recommendations to a user group that visited a specific location.

In terms of evaluation measurements, the MRR results of the proposed methods on top-10 recommendations are around 0.9, which has a large difference from the results in other measures. Considering the MRR's evaluation purpose focusing on only a single item, we can see that the proposed methods have quite high performance to recommend the next service that can be used by a target user group at a specific location. Additionally, the second high performance of proposed methods is $mNDCG$ for top-k recommendations. It means that the NNU_{DM_b} and NNU_{DB_b} using the vector similarity trained by Doc2Vec techniques work well in a graded rating fashion.

Let's discuss the results in terms of top-k recommendations (i.e., the number of recommended items). With the more decrease of k, the performance of the baseline approaches is worse, except for in mAP. The MRR shows a larger decrement than the other measurement in the baselines, while our methods have relatively smaller decrements in the measurement. The methods' MRR performance is higher than 53% on the top-2 recommendation. These results emphasize the potential capability of our methods for a next service recommendation which can be used in many recommendation purposes such as tour planning, dynamic recommendation, and so on. Interestingly, the proposed methods show slightly higher performance in the mAP when it recommends the smaller numbers of business services. Furthermore, except for MRR, the performance decrements of the proposed methods are in general smaller than those of the other approaches. These results imply that the proposed methods provide services with more proper ordering regardless of the number of recommended services than the others.

7. Conclusion

Millions of card transactions, which eventually reflect tourist consumption behaviors and patterns, generate a massive volume of big data in tourism. However, it is difficult to directly apply the available data to recommender systems since the huge amount of data contains generally implicit preferences of the tourists. Furthermore, the row data of card payment transactions, which contain personal information, may not be available in terms of GDPR. In addition to these, it is important to properly reflect a spatial factor in tourism recommender systems.

To address these challenges, we propose tourism service recommendation methods based on Doc2Vec techniques, a set of well-known methods from the natural language processing domain, for a target user group visiting a specific location. In order to model the card transaction data statistically processed to protect personal information, the techniques train a model on the service usage history of user groups along with spatial information. The vector representations are then used in three recommendation methods to make recommendations by considering the location information.

Experiments on around fourteen million statistical card transaction data demonstrated that the proposed recommendation methods outperform other baseline methods. In particular, comparing the proposed methods with other baselines emphasized the positive influences of spatial information on recommendation performance. Furthermore, these methods showed the capability to deal with various top-k recommendations without high decrements in recommendation performance than the other compared approaches. In addition to these, the proposed methods are able to recommend business services to new user groups whose data does not exist in the dataset and are directly applied to raw transaction data to provide recommendations to individuals.

Acknowledgments. This work was supported by the Ministry of Education of the Republic of Korea and the National Research Foundation of Korea (NRF-2019S1A3A2098438).

References

1. Al-Ghossein, M., Abdessalem, T., Barré, A.: Cross-domain recommendation in the hotel sector. In proceedings of the Workshop on Recommenders in Tourism, RecTour 2018, co-located with the 12th ACM Conference on Recommender Systems (RecSys 2018). Vancouver, Canada. (2018)
2. Almeida, F., Almeida, J., Mota, M.: Perceptions and trends of booking online payments in tourism. Journal of Tourism and Services, Vol. 10, No. 18, 1–15. (2019)
3. Amato, F., Moscato, V., Picariello, A., Piccialli, F.: SOS: A multimedia recommender system for online social networks. Future Generation Computer System, Vol. 93, 914–923. (2019)
4. Baek, J.W., Chung, K.Y.: Multimedia recommendation using word2vec-based social relationship mining. Multimedia Tools and Applications, 1–17. (2020)
5. Bahramian, Z., Abbaspour, R.A.: An ontology-based tourism recommender system based on spreading activation model. International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences, Vol. 40. (2015)
6. Cai, G., Lee, K., Lee, I.: Itinerary recommender system with semantic trajectory pattern mining from geo-tagged photos. Expert Systems with Applications, Vol. 94, 32–40. (2018)
7. Casillo, M., Clarizia, F., Colace, F., Lombardi, M., Pascale, F., Santaniello, D.: An approach for recommending contextualized services in e-tourism. Inf. Vol. 10, No. 5, 180. (2019)
8. Chaudhari, K., Thakkar, A.: A comprehensive survey on travel recommender systems. Archives of Computational Methods in Engineering, 1–27. (2019)
9. Chen, L., Yang, W., Li, K., Li, K.: Distributed matrix factorization based on fast optimization for implicit feedback recommendation. J. Intell. Inf. Syst., Vol. 56, No. 1, 49–72. (2021)
10. Dev, H., Hamooni, H.: Profiling US restaurants from billions of payment card transactions. In proceedings of the 7th IEEE International Conference on Data Science and Advanced Analytics, Sydney, Australia. (2020)
11. Du, M., Christensen, R., Zhang, W., Li, F.: Pcard: Personalized restaurants recommendation from card payment transaction records. In proceedings of the World Wide Web Conference, WWW, San Francisco, CA, United States. (2019)
12. Esmaeili, L., Mardani, S., Golpayegani, S.A.H., Madar, Z.Z.: A novel tourism recommender system in the context of social commerce. Expert Systems with Applications, Vol. 149, 113301. (2020)
13. Esmeli, R., Bader-El-Den, M., Abdullahi, H.: Using word2vec recommendation for improved purchase prediction. In proceedings of the 2020 International Joint Conference on Neural Networks, Glasgow, United Kingdom. (2020)
14. Guo, L., Liang, J., Zhu, Y., Luo, Y., Sun, L., Zheng, X.: Collaborative filtering recommendation based on trust and emotion. J. Intell. Inf. Syst. Vol. 53, No. 1, 113–135. (2019)

15. Hong, M.: Decrease and conquer-based parallel tensor factorization for diversity and real-time of multi-criteria recommendation. Information Sciences, Vol. 562, 259–278. (2021)
16. Hong, M., Jung, J.J.: Multi-criteria tensor model consolidating spatial and temporal information for tourism recommendation. J. Ambient Intell. Smart Environ., Vol. 13, No. 1, 5–19. (2021)
17. Hong, M., Jung, J.J.: Multi-criteria tensor model for tourism recommender systems. Expert Systems with Applications, Vol. 170, 114537. (2021)
18. Hong, M., Koo, C.M., and Chung, N.H.: DSER: Deep-Sequential Embedding for single domain Recommendation. Expert Systems with Applications, Vol. 208, 118156. (2022)
19. Kotiloglu, S., Lappas, T., Pelechrinis, K., Repoussis, P.: Personalized multi-period tour recommendations. Tourism Management, Vol. 62, 76–88. (2017)
20. Le, Q.V., Mikolov, T.: Distributed representations of sentences and documents. In proceedings of the 31th International Conference on Machine Learning, Beijing, China. (2014)
21. Li, Y., Xu, L., Tian, F., Jiang, L., Zhong, X., Chen, E.: Word embedding revisited: A new representation learning and explicit matrix factorization perspective. In proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina. (2015)
22. Lu, J., Wu, D., Mao, M., Wang, W., Zhang, G.: Recommender system application developments: A survey. Decis. Support Syst. Vol. 74, 12–32. (2015)
23. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In proceedings of the 27th Annual Conference on Neural Information Processing Systems, Lake Tahoe, Nevada, United States. (2013)
24. Musto, C., Semeraro, G., de Gemmis, M., Lops, P.: Word embedding techniques for content-based recommender systems: An empirical evaluation. In proceedings of the 9th ACM Conference on Recommender Systems, Vienna, Austria. (2015)
25. Ozsoy, M.G.: From word embeddings to item recommendation. CoRR abs/1601.01356 (2016)
26. Park, S.T., Liu, C.: A study on topic models using lda and word2vec in travel route recommendation: focus on convergence travel and tours reviews. Personal and Ubiquitous Computing, 1–17. (2020)
27. Pessemier, T.D., Dhondt, J., Martens, L.: Hybrid group recommendations for a travel service. Multim. Tools Appl. Vol. 76, No. 2, 2787–2811. (2017)
28. Rizvi, S., Kurtz, A., Williams, I., Gualdoni, J., Myzyri, I., Wheeler, M.: Protecting financial transactions through networks and point of sales. Journal of Cyber Security Technology, Vol. 4, No. 4, 211–239. (2020)
29. Sharifihosseini, A.: A case study for presenting bank recommender systems based on bon card transaction data. In proceedings of the 9th International Conference on Computer and Knowledge Engineering. (2019)
30. Zhang, S., Yao, L., Sun, A., Tay, Y.: Deep learning based recommender system: A survey and new perspectives. ACM Comput. Surv. Vol. 52, No. 1, 5:1–5:38. (2019)

Minsung Hong is a researcher in Data Science Lab (DSL) of Korea Electric Power Corporation, South Korea since June 2022 after a research professor in Kyung Hee University from February 2021 to May 2022. He was a postdoctoral researcher participating in several EU Horizon2020 and Norway national projects in Western Norway Research Institute, Norway from February 2018 to January 2021. He received the Ph.D. degree in Computer Engineering from Chung-Ang University in 2018. His research topics are recommender systems, big data, artificial intelligence, data mining, machine learning, and natural language processing.

Namho Chung is a Dean of College of Hotel & Tourism Management, Professor at the Smart Tourism Education Platform and the Director of Smart Tourism Research Center at

Kyung Hee University in Seoul, South Korea. He has been a Visiting Research Fellow at School of Hospitality and Tourism Management, University of Surrey in Guildford, UK. His research interests include travel behavior, information search and decision making, destination marketing, knowledge management. Currently, he leads smart tourism city projects in the Ministry of Education of the Republic of Korea and the National Research Foundation of Korea. His name listed in the Hall of Fame at Kyung Hee University for his outstanding achievements.

Chulmo Koo is a Professor of Smart Tourism Education Platform (STEP), College of Hotel and Tourism Management at Kyung Hee University, South Korea. He is an Editor-in-Chief of Journal of Smart Tourism and has a strong record of smart tourism research and scholarship with significant contributions to the smart tourism field.

Received: Jun 20, 2022; Accepted: December 29, 2022.