How Airbnb Tells You Will Enjoy Sunset Sailing in Barcelona? Recommendation in a Two-Sided Travel Marketplace

Liang Wu
Airbnb
San Francisco, CA
liang.wu@airbnb.com

Mihajlo Grbovic
Airbnb
San Francisco, CA
mihajlo.grbovic@airbnb.com

ABSTRACT

A two-sided travel marketplace is an E-Commerce platform where users can both host tours or activities and book them as a guest. When a new guest visits the platform, given tens of thousands of available listings, a natural question is that what kind of activities or trips are the best fit. In order to answer the question, a recommender system needs to both understand characteristics of its inventories, and to know the preferences of each individual guest. In this work, we present our efforts on building a recommender system for Airbnb Experiences, a two-sided online marketplace for tours and activities. Traditional recommender systems rely on abundant user-listing interactions. Airbnb Experiences is an emerging business where many listings and guests are new to the platform. Instead of passively waiting for data to accumulate, we propose novel approaches to identify key features of a listing and estimate guest preference with limited data availability.

In particular, we focus on **extending the knowledge graph** and **utilizing location features**. We extend the original knowledge graph to include more city-specific concepts, which enables us to better characterize inventories. In addition, since many users are new to the business, and the limited information of cold-start guests are categorical features, such as locations and destinations, we propose to utilize categorical information by employing additive submodels. Extensive experiments have been conducted and the results show the superiority of the proposed methods over state-of-the-art approaches. Results from an online A/B test prove that the deployment of the categorical feature handling method leads to statistically significant growth of conversions and revenue, which concludes to be the most influential experiment in lifting the revenue of Airbnb Experiences in 2019.

KEYWORDS

Top-N Recommendation; Knowledge Graph; A/B Testing

ACM Reference Format:
Liang Wu and Mihajlo Grbovic. 2020. How Airbnb Tells You Will Enjoy Sunset Sailing in Barcelona? Recommendation in a Two-Sided Travel Marketplace. In *43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20), July 25–30, 2020, Virtual Event, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3397271.3401444

SIGIR '20, July 25–30, 2020, Virtual Event, China

ACM ISBN 978-1-4503-8016-4/20/07…$15.00
https://doi.org/10.1145/3397271.3401444

1 INTRODUCTION

When a user visits Airbnb Experiences[1], how can we find the correct tour or activity to recommend? It is challenging since the business is still new to many of our visitors - meaning that we don't possess enough knowledge of users. Existing research on recommender systems, such as content-based models and collaborative filtering based recommendation, mainly rely on abundant amount of user and user-item interaction data to make accurate predictions. For example, content-based methods require profile or survey data to know a user's preference, and a collaborative filtering method utilizes buyers of an item to describe the item, and the purchased items to describe the buyer. As the second startup of Airbnb, a key challenge of recommendation in an emerging platform is lack of user-item interaction statistics, which hinders us from characterizing the inventories and guests.

Therefore, we aim to discover the associations between guests and listings with additional information sources when transaction data is limited. In particular, on the host side, we aim to extend the knowledge graph to include more city-specific concepts on tourism, which will help characterize a new listing rapidly. On the guest side, we propose to utilize the user travel information through the rental booking history. For example, basic profile information such as language, residency and travel destinations are mostly available on the platform of Airbnb. The available information of guests are usually categorical, and the problem is reduced to handling categorical features.

The first task is to build an extensive knowledge graph on tourism concepts. Though it might be easy to manually build the existing knowledge graph by employing a group of experts, making the knowledge graph extensive enough to include various characteristics a global marketplace may have can be challenging. For example, the existing knowledge graph focuses on concepts like "Art and Culture[2]", "Food and Drink[3]", and "Sports[4]". However, many country- and city-specific concepts are missing. Manually compiling such a list of concepts can be time-consuming and labor-intensive. Instead, we leverage the content information of existing inventories as well as search trends data of web search to automatically expand the knowledge graph. Figure 1 illustrates how a knowledge graph can influence the product design. Concepts of a knowledge graph can be used to help showcase different categories

[1] https://www.airbnb.com/s/experiences
[2] https://www.airbnb.com/s/experiences/art-and-culture
[3] https://www.airbnb.com/s/experiences/food-and-drink
[4] https://www.airbnb.com/s/experiences/sport

that are available in a city, including "Sports", "Food and Drink", *etc.* However, the categories empowered by the knowledge graph focusing on generic topics cannot cover some city-specific concepts. By extending the knowledge graph, we aim to include "Harry Potter Tours" in London and "Sunset Sailing" in Barcelona.

The second task is to leverage the limited user profile data to uncover their interests - or more concretely, the association between user profile information and listing characteristics. A straightforward method is to directly use user profile information, and treat the data as additional user features. Collaborative filtering, graphical models, or deep learning methods can then be applied by using both user-item interaction data and attributes of users. However, due to the limited amount of user-item interaction data, and the exponential increase of feature space that can be caused by categorical features, classic learning methods can be prone to overfitting. Therefore, we will focus on finding a better way to effectively utilize categorical features of user profile given limited user-item interaction information.

In a nutshell, we focus on improving the recommender system of Airbnb experiences given limited availability of user-item interaction data. A principle framework of recommendation is implemented to connect listing characteristics with user profile features. In order to improve the effectiveness of the recommender system, we investigate different approaches to discovering concepts from content information of listings to expand the knowledge graph. On the guest side, we propose a novel method to deal with categorical features in user profile data. The proposed framework has been proven to outperform existing methods significantly in both online and offline experiments. Key contributions of the work can be summarized as,

- **Recommender System on a Two-Sided Travel Platform** Though recommender systems have been extensively studied in various industries including tourism, recommendation on a two-sided travel marketplace is a relatively novel problem due to its distinct features - the transaction frequency is lower comparing to other two-sided marketplace like Ebay or Etsy, and the two-sided nature also distances itself from most business to consumer (B2C) websites, *e.g.,* inventories of a two-sided marketplace can be more complicated in terms of its categories and characteristics. We present several challenges and introduce our solutions.
- **Knowledge Graph Expansion for Tourism Applications** Classic knowledge graph research focuses on building concepts that are more generic, and domain-specific knowledge graph research rarely discusses topics related to travel and tours. By contrast, we precisely focus on the travel domain that involves tightly with locations, and extend the generic terms into more location-specific concepts.
- **Recommendation with Limited Data Availability** An additional information we can utilize is user profiles, such as travel destination and user origin. Due to lack of user-item interaction data, we find directly using the categorical information easily leads to overfitting. Instead, a novel method of dealing with categorical features has been proposed.
- **Large-Scale Online Experiments with Statistically Significant Improvement** In order to validate the proposed approaches, extensive experiments have been conducted using real-world data. The proposed method outperforms competitive baseline methods in offline evaluation. In online experiment, the proposed method achieves statistically significant improvement in terms of several key user engagement metrics, such as bookings and revenue.

The rest of the work is organized as follows: we present the motivation behind the work and define the computational problems we will be focusing on in Section 2. The proposed methods will be introduced in Section 3. In Section 4, we will present the extensive experiments we conducted to validate the proposed methods in both online and offline environment. Section 5 lists how the proposed approaches relate to existing techniques. We conclude the work and point out future directions in Section 6.

2 PROBLEM STATEMENT

Table 1: Notations and Description

Notation	Description
$\mathbf{P}$	Set of listings
$\mathbf{U}$	Set of guest users
$\mathbf{Q}$	Set of search queries
$\mathbf{T}$	Set of transactions between users and listings
$\mathbf{C}$	Content information of listings
$\mathbf{F}$	Profile information of guest users
w_p	A word appearing in content of listing p
$\mathbf{L}$	Set of locations
$PP(w,l)$, $NPP(w,l)$	Search trends popularity and normalized popularity of a keyword for a certain location
$BK(w,l)$, $NBK(w,l)$	Bookability and normalized bookability of a keyword for a certain location

As discussed above, we focus on improving the recommender system of a two-sided travel marketplace, Airbnb Experiences. Key challenges we face include cold-start problems - we lack enough knowledge of our inventories and users, and we there is also a lack of interaction data between users and listings, on which a recommender system would mostly rely. In order to improve the recommendation accuracy when data is sparse, we propose to enhance the understanding of inventories by extending the knowledge graph to be more comprehensive in tourism concepts, and investigate how profile features that new users possess can be utilized to reveal user interests. Computationally, the tasks we will work on are outlined below,

Definition 2.1. Listing recommendation in travel marketplace Given a set of listings $p \in \mathbf{P}$, a candidate guest user $u \in \mathbf{U}$, and transaction history of the platform $(p,u) \in \mathbf{T}$, we try to recommend the listings to a user that is most likely to lead to a booking.

The definition above describes a classic recommender system problem where the listings here are tours and activities instead of

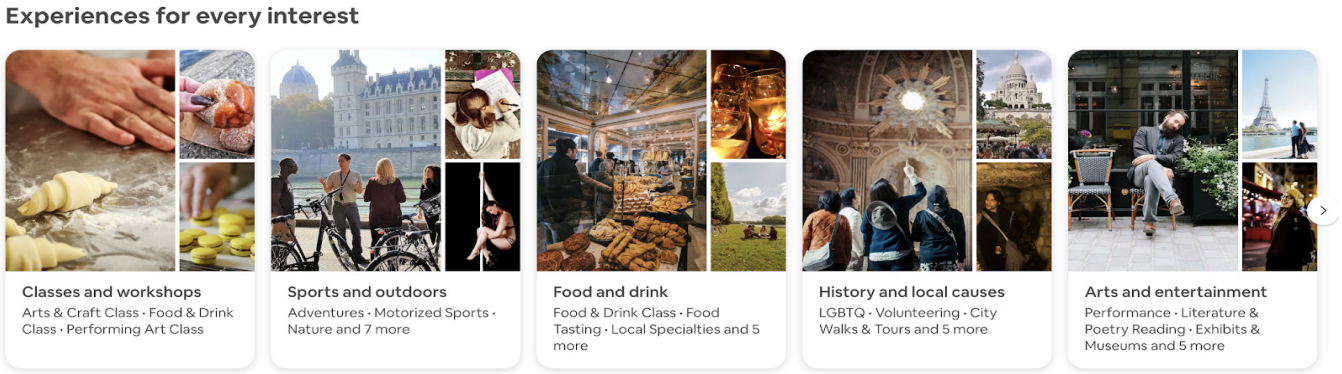


Figure 1: Listing categories using the original knowledge graph. Concepts like "Sports" and "Food and Drink" are available. We aim to extend the generic concepts into more city-specific concepts.

product or movies. We further decompose the problem into two key subtasks to tackle the challenges of the business.

Definition 2.2. Knowledge graph expansion to include travel specific concepts Given a set of listings $p \in \mathbf{P}$ and its content information extracted from titles and descriptions $c_p \in \mathbf{C}$, we aim to expand the concepts of the knowledge graph by collecting keywords and phrases from $\mathbf{C}$.

Definition 2.3. Incorporating categorical features in user profiles Given a set of users $u \in \mathbf{U}$ and its categorical features from profile data $f_u \in \mathbf{F}$, we aim to utilize features to improve the accuracy of the problem in Definition 2.1.

3 RECOMMENDATION IN AIRBNB EXPERIENCES

3.1 Knowledge Graph Expansion

First we will introduce how the knowledge graph can be expanded to be more comprehensive on tourism concepts. The existing knowledge graph consists of a tree-structure taxonomy covering most generic concepts of tourism. We illustrate parts of the original knowledge graph in Figure 2, where the concepts are structured like a tree and the top nodes represent concepts that are more generic. For example, the top level concept can be "Entertainment" or "Sports", and more specific concepts such as "Comedy" and "Dance" reside below "Entertainment". The second-level concept "Dance" can be further decomposed into "Ballet", "Disco" and so on.

A knowledge graph can be used to tag listings with concepts in the taxonomy. For example, a new listing that is tagged with "Ballet" can be quickly categorized into "Dancing" and "Sports". When we recommend "Other Dancing Activities You May Get Interested", the listing will naturally be considered for being highlighted.

A knowledge graph that is built by a group of experts usually has nice structure and it can be easily extended. A key drawback is that it may fail to be comprehensive when various new listings are pouring in. For example, a key travel concept in Beijing is Hutong, a type of streets or alley that are common in northern Chinese cities. Without further expanding the knowledge graph, the Hutong listings will have to be tagged with "City Tour". It will

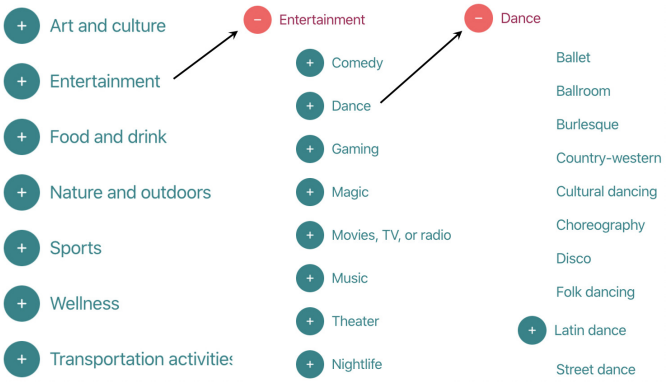


Figure 2: Parts of the original knowledge graph where the taxonomy is in a tree structure. We show three levels of concepts where the top levels are more generic concepts.

be frustrating if we show a lot of "City Tour" listings when a user is particularly interested in "Hutong". Some more examples include "Harry Potter" in London and Edinburgh, and "Medici" in Florence.

Therefore, we aim to improve the knowledge graph by including more of these extended keywords. The basic idea is to discover concepts from content information $\mathbf{C}$ of listings, such as keywords in titles and descriptions. There are mainly two ways of detecting keywords from natural language, *i.e.,* supervised and unsupervised. Since this work focuses on extending such specific keywords from scratch, meaning that there is a lack of enough annotated data, we adopt unsupervised methods to automatically discover keywords.

We first investigate existing methods such as TF-IDF. In particular, we regard each listing as a single document c_p and for each word w_i, we calculate the Term Frequency (TF) as $|w_i \in c_p|$, and Inverse Document Frequency (IDF) as $log \frac{|\mathbf{P}|}{|\{p|w_i \in c_p\}|}$. This is a classic unsupervised method and a problem is that listings descriptions are very different from traditional documents. Given relatively short description while a large number of listings, TF-IDF can fail to capture the keywords.

Figure 3: An example of listings on Airbnb Experiences, including a cover photo, a title and a short description.

We then investigate using existing approaches such as rule-based methods and graph-based methods. For example, we test using KP-Miner[9] to directly identify keywords using rules. However, the rules that work fine on normal documents may not be applicable for short descriptions of the listings. An example of the listing is depicted in Figure 3.

Another widely used approach is keyword ranking - ranking top keywords that are most representative to a document. For example, the documents can be regarded as graph where each word can be viewed as a vertex. By building edges between words using occurrences, graph mining methods such as PageRank can be used to estimate the importance of a word [18]. Several subsequent works along this stream improve the approach by introducing global word distribution information [26] and topic distribution information [4]. A key challenge here is still the short description and high diversity of listings make it difficult for classic methods to work.

The scarcity of text data motivates us to leverage external information sources. Motivated by recent advances in representation learning, we next investigate utilizing pretrained embedding models. Since pretrained text embedding models, such as Word2Vec [19] and Sent2Vec [22], are built upon external large corpus, adopting them can help alleviate the challenges brought by lack of content information. The pretrained embedding models can map words into a embedding vector with moderate dimensionality. Word-word similarity can directly be measured using the cosine distance between vectors. In this work, we may represent each word using a embedding vector. However, a limitation is that word embeddings can only represent single words, while many keywords in our case, such as "Harry Potter", are phrases consisting of multiple words.

Sent2Vec generate embeddings for sentences, where a sentence is decomposed into n-gram features. Therefore, a phrase can similarly represented by an embedding vector. Using the same method, an entire paragraph can also be embedded into a vector, as the description of the listings are relatively short. Since a keyphrase is more likely to be correlated to the document, we can use the embedding vectors to rank all keyphrases based on the cosine similarity. The above process is similar to that of text ranking [3].

3.1.1 Top Keyphrase Selection. The above process focuses on ranking keyphrases from a listing's title and description. However, the main focus of this work is to expand the knowledge graph using keyphrases from listings. Based on the lessons we learn from multiple launches of different knowledge graph concepts, an ideal concept for knowledge graph expansion has following features: (1) Corresponds to enough inventories: the concept should cover a category of inventories instead of a single listing since it will be used to recommend listings under the concept; (2) Location-specific: the concepts we focus for expansion are highly specific to a certain location, such as "Hutong" in Beijing and "Harry Potter" in London; (3) Known by guest users: though unavailable in the existing knowledge graph, the concepts should be known by the guest users, thus being recommendable; (4) Listings with the keyword are bookable: it is also appealing that the listings under a new concept have already been well booked by guests, meaning that the new node will be promising in the recommender system.

Therefore, we propose several new metrics to help rank concepts for merging into the knowledge graph. In order to capture how many listings a keyphrase corresponds to, we define a metric called Listing Frequency (LF) as,

$$LF(w) = |\{p|w \in c_p\}|,$$

where we calculate the number of listings that a keyword has appeared in. A popular keyphrase is more likely to be a tourism concept. Next, we introduce a metric to describe how location-specific a keyphrase is. Motivated by the formulation of Inverse Document Frequency, we will introduce the metric of Inverse Location Frequency (ILF) as,

$$ILF(w) = |log\frac{|\mathbf{L}|}{|\{l|w \in c^l\}|}|,$$

where we use c^l to denote content information of a location's listings altogether. Similarly to IDF, ILF measures how unique and important a keyphrase candidate is to a specific location. Next, we will propose a metric to measure the extent of how well known a keyword is to guests.

By examining the search log data, we can collect all queries users search in a specific market. Our assumption is that the searched keywords can represent how well a keyword is known to the guests. Therefore, we propose a metric named search trends popularity as,

$$PP(w, l) = |\{q \in \mathbf{Q}^l|w \in q\}|,$$

where we use $\mathbf{Q}^l$ to denote the subset of search queries for a location l. $PP(w, l)$ represents the raw count of how many queries in a location containing the word w. By normalizing the raw count with the most popular word in a location, we can get its normalized value NPP,

$$NPP(w, l) = \frac{PP(w, l)}{\max_{w' \in \mathbf{W}, l'=l} PP(w', l')},$$

where NPP can directly measures how well-known a keyphrase is to a particular location. We notice that the user traffic comes from both native search engines and external search engines. Therefore, the popularity metric $PP(\cdot)$ considers both local traffic using Airbnb and external search trends data, such as Google Trends[5].

Next, we try to denote how bookable a candidate concept can be, and we propose the notion of bookability,

$$BK(w, l) = |\{(p, u) \in \mathbf{T}^l|w \in c_p\}|,$$

[5]https://trends.google.com/trends/

where we measure bookability by calculating the number of transactions consisting of listings with the certain keyword in a certain location. Similarly, we can also have a normalized bookability metric NBK.

In order to implement the metrics, we calculate all statistics using a 90-day window, which allows us to find newly emerging concepts as well as old ones by using historical data. The popularity and bookability metrics are both specific to a certain location - we aim to find top keyphrases on a location level. In this work, the location granularity is city.

By using the four above metrics, we rank all candidate keyphrases to generate promising new concepts for the knowledge graph. To conclude, there are two main steps of the framework, *i.e.,* producing candidate keyphrases based on listing titles and descriptions, and ranking the candidates based on rules we compiled. We outline the overall algorithm in Algorithm 1.

Algorithm 1 Algorithm of Knowledge Graph Expansion

Input: $\mathbf{P}, \mathbf{Q}, \mathbf{T}, \mathbf{C}, \mathbf{L}$, budget of expansion for a location: k
Output: Top k candidates for a location
1: For each listing $p \in \mathbf{P}$:
2: Generate candidate keyphrases $\mathbf{W'}$
3: For each word $w \in \mathbf{W'}$:
4: Estimate $LF(w)$, $ILF(w)$
5: For each location $l \in \mathbf{L}$:
6: For each word in $w \in \mathbf{W'}$:
7: Estimate $NPP(w, l)$, $NBK(w, l)$
8: Calculate ranking score as $LF \times ILF \times NPP \times NBK$
9: Return top k candidates in a location

Step 1-2 aim to find candidate keyphrases from content information of listings. Step 3-4 ensure that the candidates represent location-specific keyphrases that have enough underlying inventories. Step 5-7 aim to find candidates that are well known to users and also represent bookable listings on the website. We use ranking scores in Step 8 to select top k candidate keyphrases in each location. We will present experimental results to show effectiveness of different unsupervised keyword extraction methods in experiments.

3.2 Categorical Features in Profiles Handling

In this section, we focus on utilizing the categorical features from user profiles. Since user location is the most important categorical feature for travel recommendation in our case, we will use the feature as an example for rest of the work. The location here is on the city level. We will use location and city interchangeably.

A straightforward way to directly employ the categorical features, or directly encode it as a binary feature. For example, if the location feature has 200 cities, then the feature will be decomposed to 200 binary features. In our studies the methods significantly downgrade the model performance. Since our data scale is relatively small, the rapid increase of feature space, especially features of high sparsity, easily leads to overfitting.

Motivated by deep learning research, another way is to transform categorical features into embeddings [33]. A practical challenge is that the deep learning architecture also requires large amount of data. We test a classic neural recommendation model and the results will be reported in the experiment section.

In this work, a novel method that has not been well studied before is to use categorical features to build separate models. The idea originates from the question that if users from different countries behave very differently, should we just build a separate model for each of them? We select top 30 guest countries, and build a separate model for each of them. The recommendation accuracy of 30 separate models is much lower than a global model, which is not surprising since each model will have much less data.

In order to alleviate the challenge of small data, we investigate another method of global-local model. We train a global model using all data while each top user origin country will have a submodel that is trained using only the local data. We name the model LocalBoost and the change brings about significant improvement over the existing system. LocalBoost is a methodology that can be realized using different learning algorithms. We will introduce the implementation details with two specific models, *i.e.,* neural recommender systems, and gradient boosting [10].

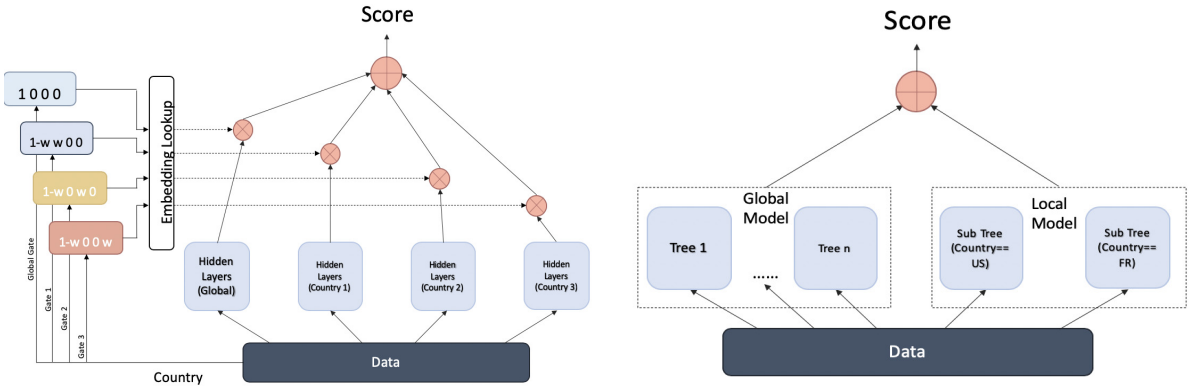
Figure 4 illustrates the two model structures. We pick neural recommendation [11] since deep neural has been trending recently, and it has been reported in several recent works that a neural ranking model outperforms the state-of-the-art approaches. We are also aware of the debate about whether neural recommendation has been really outperforming traditional methods. Therefore, we also pick a shallow model here. The implementation of the shallow model is based on XGBoost [6], which has been the best single model in various data competitions.

Figure 4(a) is the LocalBoost with deep neural networks. We assume there are three different user origin countries to be boosted in this case as an example. There are two kinds of hidden layers, where the first one is a global model and the next three are local models for specific countries. We introduce four gate embeddings here to control the turning on of the different models. For example, all data instances will go through the first hidden layers, while only data from "Country 1" will go through the second. The framework prevents countries from overfitting using only local data.

Figure 4(b) is the LocalBoost with gradient boosting. We assume there are two different user origin countries to be boosted in this case as an example, *i.e.,* United States (US) and France (FR). There are two kinds of decision trees here, where the first bunch represents a global model and the next two are local models for US and FR, respectively. The results of all trees are gathered together to make a prediction. For example, a guest not from US or FR will get recommendation results based only on the global model, while the recommendation for a French guest will be determined jointly by the global model and the FR sub model. The framework prevents countries with less data from overfitting using only local data.

3.3 Discussion

To conclude, the proposed framework concludes two main components to improve the recommender system. On the host side, we propose to expand concepts of taxonomy for the knowledge graph, extending global and generic concepts such as "Food and Drink" and "Sports" to city-specific concepts such as "Mezcal Tasting" and



(a) LocalBoost model with a neural recommendation framework.

(b) LocalBoost model with a gradient boosting framework.

Figure 4: The LocalBoost framework implemented with different learning algorithms.

"Lucha Libre" in Mexico City. On the guest side, we propose a global-local model to deal with categorical features so as to cope with the challenges of cold-start and lack of user-item interaction data. In the next section, we will introduce how we conduct experiments to validate the proposed approaches.

4 EXPERIMENT

In this section, we will conduct experiments to validate the effectiveness of the proposed methods. In particular, we will try to answer the following questions through offline and online experiments:

- What additional concepts can the proposed knowledge graph expansion algorithm select to expand?
- How effective is the LocalBoost utilizing categorical features comparing with other competitive baselines?
- How well is the proposed framework in lifting key user engagement metrics such as bookings and revenue?

In order to answer the questions, we conduct experiments using real-world data from Airbnb. We conduct experiments differently on host- and guest-side. Since it is difficult to evaluate the knowledge graph extension algorithm, we will focus on case studies to explain the additional concepts the proposed algorithm can detect. On the guest side, we conduct both offline and online experiments to measure its effectiveness in utilizing location features. We bring the proposed location feature handling framework into online traffic and conduct an online A/B testing for over 3 weeks.

4.1 Host-side Experiment on Knowledge Graph Expansion

4.1.1 Data. We randomly sample 40,000 active listings and collect the publicly available content information including titles and descriptions. The title and description are manually compiled by the host of a listing. There are over 30 thousand hosts involved, from 20 countries and 93 cities. The average number of words in titles is 7, and the average length of description is 83, leading to an average length of 90 words for each listing.

Table 2: Precision of different methods on extracting keyphrases.

Method	k=1	k=2	k=3	k=4	k=5
TFIDF	0.40	0.34	0.37	0.29	0.27
KPMiner	0.61	0.61	0.55	0.58	0.52
TextRank	0.61	0.60	0.48	0.44	0.48
CollabRank	0.64	0.62	0.54	0.43	0.41
TopicRank	0.56	0.61	0.49	0.42	0.43
EmbedRank	0.65	0.62	0.51	0.43	0.44

4.1.2 Effectiveness of Keyphrase Extraction. In the first experiment, we focus on evaluating the keyphrases extracted by different methods. Note that these are the phrases extracted from listings, and they are candidates for the next phase of knowledge graph expansion. We introduce several competitive baselines here.

- TFIDF is a classic keyword extraction method that is based on term frequency and inverse document frequency. We apply TFIDF by regarding each listing as a document.
- KPMiner [9] is a rule-based keyword extraction method that has been proven effective in TREC. We include KPMiner to test the effectiveness of rule-based methods.
- TextRank [18] is a graph-based method that ranks words in a document in a graph mining manner. Each word is regarded as a vertex and the centrality of a node can be interpreted as the importance of a keyword for the document.
- CollabRank [26] extends TextRank by utilizing clustering of documents. Document clusters provide more global information of the word distribution and alleviate the information scarcity issue.
- TopicRank [4] further extends TextRank by using latent topic modeling. Topic modeling can be regarded as soft clustering of documents and it provides more global word distribution information.
- EmbedRank [3] is also built upon the idea of borrowing global information to extend TextRank, but it leverages

the word and document embeddings instead of document clustering or topic modeling.

4.1.3 Results of Keyphrase Extraction. In order to evaluate the results of different models, we vary the expansion budget k to select top candidates for each location. Based on the results we obtained from various algorithms, a list of key concepts have been adopted to be added to the knowledge graph. Since there is lack of availability of ground truth, we utilize the finally selected candidates for evaluation. After being reviewed by a group of engineers, some candidate concepts generated by the proposed method are selected to be added to the knowledge graph. We regard the selected list as the ground truth, and we evaluate different methods with a varying budget k. All results with different budget k are shown in Table 2. The results are mostly helpful for comparing different baseline methods, while it can be unfair to compare the baselines with the proposed method. Therefore, we will show more case study results later in this subsection.

According to the results in Table 2, The runner-up method is EmbedRank which also uses pretrained word embeddings to alleviate content information sparsity. An interesting observation is that the models seem sensitive to the selection of k. This may be caused by the different scales of markets - a large market may have more city-specific keyphrases while a smaller one may have less. Therefore, we will investigate the problem of finding the best k for a certain market in the future. An interesting finding is that different methods may share many results. However, the overlap between the proposed method and other baseline methods is slim. We randomly pick some cities to show the candidate concepts generated only by the proposed method in Figure 5. Some of the results are very well known, but recently emerging as a popular concept among the listings, such as "Football" in Buenos Aires. Many concepts are also closely related to a location, such as "Tuk-tuk" and "Tuscan Wine". "Sunset Sailing" in Barcelona can help the further specify listings that have been tagged with the more generic concept of "Sailing".

4.2 Guest-side Experiment on Categorical Feature Handling

In this subsection, we will present experimental results on guest-side methods.

4.2.1 Data. In this experiment, we obtain user browsing and booking data to build an offline dataset. We randomly subsample data to build a dataset with both logged-in and logged-out-users.

We employ two datasets on logged-in users and logged-out users since their browsing behaviors are intrinsically different. Logged-in users usually have booking history on Airbnb and are more likely to know about Airbnb experiences. More importantly, the profile features including guest origin country, on which we focus in this work, are more accurate. We can directly obtain user origin country for all logged-in users. In order to obtain the data for logged-out users, we use the query country of the search request.

4.2.2 Effectiveness of Categorical Feature Handling. As discussed above, we will investigate different ways of handling categorical features. The methods that are used in the experiment include:

- Binary Coding GBDT (BC-GBDT): We directly use the categorical features in the model by transforming them into multiple binary features. As discussed above, it would lead to rapid increase of dimensionality. We apply this on both the neural recommendation model [11] and the GBDT model [6].
- Embedding NN (E-NN): When a neural recommender system is employed, categorical features can be naturally embedded as a vector of continuous features.
- Multiple Models GBDT (MM-GBDT): We split the dataset according to different user countries and an independent model will be produced for a certain country.
- Multiple Models NN (MM-NN): Similarly, we train multiple neural recommendation models based on guest origin countries for comparisons.
- LocalBoost NN (L-NN): The proposed method with Neural Network as the learning method.
- LocalBoost GBDT (L-GBDT): The proposed method with GBDT as the learning method.

We conduct the experiments in a Top-N Recommendation manner, where candidate listings are ranked using the model output score. Conventionally, we adopt the metric of NDCG@K as the evaluation metric. Normalized Discounted Cumulative Gain (NDCG) is a widely used of ranking quality, and K is the top K positions we consider for recommendation. We test various positions and show the results for K=3 and K=5 in this study due for brevity. The offline results using different K values doesn't lead to inconsistent results for both logged-in and logged-out users. In our task, the positive label is "booking", which is a binary value. Therefore, we also adopt Area Under Curve (AUC) of ROC to measure the effectiveness.

The experimental results are shown in Table 3 for logged-in users, and Table 4 for logged-out users. Based on the experimental results, we can draw several observations. The proposed method achieves the best performance for both tasks; Using multiple models and using deep neural networks models lead to rapid downgrade of model performance since the data is relatively sparse. The LocalBoost framework helps improve the performance of both models in handling categorical features.

4.3 Online A/B Testing

In the online experiment, we aim to evaluate how the proposed categorical feature handling method can improve the recommendation accuracy after it is launched online. Note that the proposed method uses both expanded knowledge graph concepts and the categorical feature handling method. In particular, we deploy the proposed algorithm globally as the treatment group, and randomly assign 50% traffic to the treatment group. During the 3-week online A/B testing, we focus on the several key metrics,

- **Revenue** refers to a certain portion of user booking values that is revenue to the platform.
- **Booking** refers to the quantity of bookings of seats which is one of the most important user engagement signal.
- **Booker** refers to the quantity of users who book an item on Airbnb and it reflects how many users find search results match their search intents.
- **NU Booking** refers to the number of bookings by New Users (NU). This is an important metric measuring how a ranking algorithm escalates the growth of a marketplace.

Paris	Eiffel Tower	Louvre Museum	Le Marais	Montmartre	French Wine	Latin Quarter	French Pastry
Kyoto	Fushimi Inari	Kimono	Zen	Sushi	Sake	Gion	Arashiyama
Rio de Janeiro	Sugarloaf	Santa Teresa	Tijuca National Park	Ipanema Beach	Samba	Copacabana	Lapa
Havana	Classic Car	Viñales Valley	Cuban Rum	Cuban History	Cuban Cigars	Salsa	Rumba
Buenos Aires	Palermo	Football	Asado	San Telmo	Tango	Boca	
Bangkok	Thai Cooking	Tuk-tuk	Floating Market	Temples	Canal	Chinatown	
Madrid	Retiro Park	Tapas Tour	Flamenco	Spanish Wines	Spanish Cuisine		
Florence	Tuscan Wine	Renaissance	Uffizi Gallery	Medici			
Barcelona	La Sagrada Familia	Sunset Sailing	Tapas Tour	Paella			

Figure 5: Some example knowledge graph concepts that are only produced by the proposed method, where each row represents a city and its top candidate concepts.

Table 3: Experimental results on handling categorical features for logged-in users.

Method	NDCG@3			NDCG@5			AUC		
	Train	Vali	Test	Train	Vali	Test	Train	Vali	Test
BC-GBDT	0.3942	0.4090	0.4015	0.4181	0.3956	0.4483	0.8716	0.8497	0.8205
E-NN	0.3692	0.3592	0.3296	0.4268	0.4423	0.4459	0.8582	0.8568	0.8360
MM-GBDT	0.3461	0.3207	0.3500	0.3644	0.4423	0.4038	0.7947	0.7585	0.7614
MM-NN	0.3628	0.3185	0.3260	0.3921	0.3465	0.3129	0.7780	0.7089	0.7150
L-NN	0.3801	0.3754	0.4044	0.3895	0.4045	0.4141	0.8506	0.8416	0.8234
L-GBDT	0.3853	0.4291	0.4233	0.4493	0.4782	0.4669	0.8240	0.8000	0.8969

Table 4: Experimental results on handling categorical features for logged-out users.

Method	NDCG@3			NDCG@5			AUC		
	Train	Vali	Test	Train	Vali	Test	Train	Vali	Test
BC-GBDT	0.3771	0.3893	0.3793	0.3942	0.3901	0.4156	0.8209	0.8361	0.7651
E-NN	0.3585	0.3311	0.3154	0.4100	0.4135	0.4307	0.8454	0.7920	0.8349
MM-GBDT	0.3391	0.3153	0.3310	0.3382	0.4144	0.3883	0.7883	0.7463	0.7215
MM-NN	0.3511	0.3105	0.3196	0.3619	0.3249	0.3044	0.7342	0.6972	0.6747
L-NN	0.3665	0.3543	0.3747	0.3679	0.4031	0.3949	0.8106	0.8229	0.8172
L-GBDT	0.3673	0.4121	0.4054	0.4427	0.4584	0.4489	0.8068	0.7782	0.8654

Table 5: Online A/B testing results. We show the percentage of improvement on different metrics of the treatment group.

	Revenue	Booking	Booker	NU Booking	New Booker	See Dates	Choose Dates	Wishlist
Treatment	↑ 3.9%*	↑ 2.5%*	↑ 1.2%	↑ 1.0%	1.0%	1.2%	↑ 1.2%	↑ 3.4%*

* indicates the statistical significance level of 0.05.

- **New Booker** refers to the number of users who make their first booking. This is an important metric indicating long-term growth of an online marketplace.
- **See Dates** refers to the total number of See Dates behaviors. If users are interested in booking an item on Airbnb, they will need to see the available dates of a listing to check the availability before reserving a booking. Therefore, the number of See-Dates behaviors can indicate the number of users who find an item that meets their requirements.
- **Choose Dates** refers to the total number of Choose-Dates behaviors. The behavior reflects a user's intent to book after seeing available dates of an item.

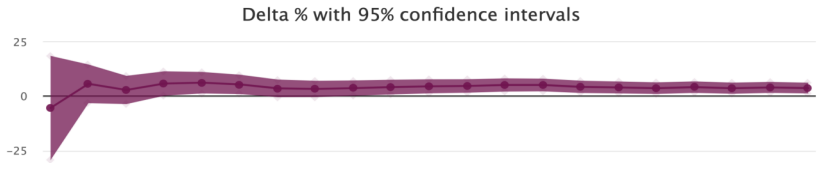


Figure 6: The percentage of change (delta) of the treatment group over the control group overtime. 95% confidence interval is also attached. Every dot on the x-axis is a day, and y-axis denotes the percentage of change.

- **Wishlist (Web)** refers to the total number of the users who add an item to the wish list. A wish list is a collection of users' favorite items. A user may save a listing for future booking or comparisons and thus it is a positive signal that reflects a listing matches a user's search intent. The reported results are based on web browser users and the results are neutral and trending negative on mobile.

The online A/B testing results are shown in Table 5. All key metrics have been lifted in the treatment group. In particular, the key metric Revenue has been significantly improved by 3.9% globally. This experiment ends up being the most influential single experiment in improving revenue and bookings in 2019. We also show how the proposed method improves the recommendation accuracy over time in Figure 6, where we can see the proposed method clearly outperforms the production system in a statistically significant manner.

5 RELATED WORK

This work aims to improve the recommendation accuracy of Airbnb Experiences. We have made efforts on both the guest side to improve the effectiveness of the recommendation model, and the host side to extend the existing knowledge graph to more city-specific concepts.

The guest-side work is related to recommendation and personalized search [8, 13, 14]. Recommender systems aim to identify interesting items for a user based on content and previous transactions [32]. Our work follows collaborative filtering methods [23], where we assume users booking similar listings have similar preferences. The proposed model focuses on dealing with categorical features. Classic solutions include binary coding the feature and learning an embedding in a neural method. The proposed method is similar to personalization methods that adjust models for different users [12, 15]. Our work can be viewed as building an additional model and establishing personalization on the location level. A similar stream of research is multi-task learning, where a global model can be adapted for to user specific ones, and the global model can be a linear ranking model [27], a collaborative filtering model [1, 21], and a decision tree-based ranking model [5]. However, the main feature that distances the proposed method from multi-task learning is that multi-task learning methods focus on a global optimization problem [29], while the proposed submodel can be optimized independently for a specific categorical feature. Building a local model in addition to the global model has been investigated in dealing with inaccurate labels [20, 31] and noisy data [30] in social networks. We investigate different learning approaches for the task, and the proposed personalized ranking models can be built upon a deep learning framework [11] or a gradient boosting framework [6].

The host-side work is related to keyword extraction and automatic knowledge graph construction. The task of keyphrase extraction is to identify important words and phrases from natural language text. Both supervised and unsupervised methods have been studied in this area. Classic supervised models reduce the task to binary classification. For example, a classification method such as Naive Bayes classifiers can be used on some handcrafted features to identify keyphrases from words in a documet [28]. In recent years, neural network-based approaches have been widely applied in this area, which are usually built upon a encoder-decoder architecture [7, 24]. Some more recent work investigate the usage of adopting attention mechanishm within a neural architecture [2, 16]. Some other methods that can be also used in a neural framework include the copying mechanism [17] that focuses on keyphrases that rarely occur in a document, the coverage mechanism that deals with the skewness of topic distributuons to include more topics, and the review mechanism to avoid including duplicate keyphrases [25]. In this work, we focus on more generic methods and investigate different methods including rule-based [9], graph-based [18, 26], topic-based [4] and embbedding-based [3], and propose a novel framework for finding potential knowledge graph concepts.

6 CONCLUSION AND FUTURE WORK

In this work, we concentrate on improving a recommender system of a two-sided travel marketplace. In order to allow for the model to better understand guest users and listings, we investigate various methods that are particularly effective for an emerging platform without abundant user-item interaction data. We propose a principle framework to expands knowledge graph using listing content and additional information sources, and propose a novel method to handle location features. The superiority of the proposed methods over competitive baselines have been proven in both online and offline experiments. It will be interesting to investigate several hyper-parameter tuning problems, such as learning the optimal knowledge graph expanding budget and the subtree structures.

ACKNOWLEDGMENTS

We would like to thank the entire Experiences Team and Relevance Team for their contributions to the projects, especially Eric Wu, and Chun How Tan. We would like to thank Dylan Hurd, Jimmy Zhuang, Bo Yu, and Chun How Tan for reviewing the keyphrase extraction results for adding to the knowledge graph.

REFERENCES

[1] Xavier Amatriain, Neal Lathia, Josep M Pujol, Haewoon Kwak, and Nuria Oliver. 2009. The wisdom of the few: a collaborative filtering approach based on expert opinions from the web. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*. 532–539.

[2] D Bahdanau, K Cho, and Y Bengio. 2019. Neural machine translation by jointly learning to align and translate. Comput Sci. 2014. *arXiv preprint arXiv:1409.0473* 9 (2019).

[3] Kamil Bennani-Smires, Claudiu Musat, Andreea Hossmann, Michael Baeriswyl, and Martin Jaggi. 2018. Simple unsupervised keyphrase extraction using sentence embeddings. *arXiv preprint arXiv:1801.04470* (2018).

[4] Adrien Bougouin, Florian Boudin, and Béatrice Daille. 2013. Topicrank: Graph-based topic ranking for keyphrase extraction.

[5] Olivier Chapelle, Pannagadatta Shivaswamy, Srinivas Vadrevu, Kilian Weinberger, Ya Zhang, and Belle Tseng. 2011. Boosted multi-task learning. *Machine learning* 85, 1-2 (2011), 149–173.

[6] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 785–794.

[7] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).

[8] Mukund Deshpande and George Karypis. 2004. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems (TOIS)* 22, 1 (2004), 143–177.

[9] Samhaa R El-Beltagy and Ahmed Rafea. 2010. Kp-miner: Participation in semeval-2. In *Proceedings of the 5th international workshop on semantic evaluation*. 190–193.

[10] Jerome H Friedman. 2002. Stochastic gradient boosting. *Computational statistics & data analysis* 38, 4 (2002), 367–378.

[11] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*. 173–182.

[12] Roy Levin, Hassan Abassi, and Uzi Cohen. 2016. Guided walk: A scalable recommendation algorithm for complex heterogeneous social networks. In *Proceedings of the 10th ACM Conference on Recommender Systems*. 293–300.

[13] Jundong Li, Liang Wu, Harsh Dani, and Huan Liu. 2018. Unsupervised personalized feature selection. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[14] Jundong Li, Liang Wu, Osmar R Zaïane, and Huan Liu. 2017. Toward personalized relational learning. In *Proceedings of the 2017 SIAM International Conference on Data Mining*. SIAM, 444–452.

[15] Peter Lofgren, Siddhartha Banerjee, and Ashish Goel. 2016. Personalized pagerank estimation and search: A bidirectional approach. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*. 163–172.

[16] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025* (2015).

[17] Rui Meng, Sanqiang Zhao, Shuguang Han, Daqing He, Peter Brusilovsky, and Yu Chi. 2017. Deep keyphrase generation. *arXiv preprint arXiv:1704.06879* (2017).

[18] Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*. 404–411.

[19] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[20] Fred Morstatter, Liang Wu, Tahora H Nazer, Kathleen M Carley, and Huan Liu. 2016. A new approach to bot detection: striking the balance between precision and recall. In *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE, 533–540.

[21] Xia Ning and George Karypis. 2010. Multi-task learning for recommender system. In *Proceedings of 2nd Asian Conference on Machine Learning*. 269–284.

[22] Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. 2017. Unsupervised learning of sentence embeddings using compositional n-gram features. *arXiv preprint arXiv:1703.02507* (2017).

[23] Yue Shi, Martha Larson, and Alan Hanjalic. 2014. Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. *ACM Computing Surveys (CSUR)* 47, 1 (2014), 1–45.

[24] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*. 3104–3112.

[25] Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling coverage for neural machine translation. *arXiv preprint arXiv:1601.04811* (2016).

[26] Xiaojun Wan and Jianguo Xiao. 2008. CollabRank: towards a collaborative approach to single-document keyphrase extraction. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*. 969–976.

[27] Hongning Wang, Xiaodong He, Ming-Wei Chang, Yang Song, Ryen W White, and Wei Chu. 2013. Personalized ranking model adaptation for web search. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*. 323–332.

[28] Ian H Witten, Gordon W Paynter, Eibe Frank, Carl Gutwin, and Craig G Nevill-Manning. 2005. Kea: Practical automated keyphrase extraction. In *Design and Usability of Digital Libraries: Case Studies in the Asia Pacific*. IGI global, 129–152.

[29] Liang Wu, Diane Hu, Liangjie Hong, and Huan Liu. 2018. Turning clicks into purchases: Revenue optimization for product search in e-commerce. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 365–374.

[30] Liang Wu and Huan Liu. 2018. Tracing fake-news footprints: Characterizing social media messages by how they propagate. In *Proceedings of the eleventh ACM international conference on Web Search and Data Mining*. 637–645.

[31] Liang Wu, Fred Morstatter, Kathleen M Carley, and Huan Liu. 2019. Misinformation in social media: Definition, manipulation, and detection. *ACM SIGKDD Explorations Newsletter* 21, 2 (2019), 80–90.

[32] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys (CSUR)* 52, 1 (2019), 1–38.

[33] Weinan Zhang, Tianming Du, and Jun Wang. 2016. Deep learning over multi-field categorical data. In *European conference on information retrieval*. Springer, 45–57.