TravelWithFriends: a Hybrid Group Recommender System for Travel Destinations

Toon De Pessemier
iMinds-Ghent University
G. Crommenlaan 8 / 201
B-9050 Ghent, Belgium
toon.depessemier@ugent.be

Jeroen Dhondt
iMinds-Ghent University
G. Crommenlaan 8 / 201
B-9050 Ghent, Belgium
jeroen.dhondt@ugent.be

Kris Vanhecke
iMinds-Ghent University
G. Crommenlaan 8 / 201
B-9050 Ghent, Belgium
kris.vanhecke@ugent.be

Luc Martens
iMinds-Ghent University
G. Crommenlaan 8 / 201
B-9050 Ghent, Belgium
luc1.martens@ugent.be

ABSTRACT

Recommender systems have proven their usefulness in many classical domains such as movies, books, and music to help users to overcome the information overload problem. But also in more challenging fields, such as tourism, recommender systems can act as a supporting tool for decision making when planning a trip. This paper proposes such a system providing group recommendations for travel destinations based on the users' rating profile, personal interests, and specific demands for their next destination. The proposed solution follows a hybrid approach, combining content-based, collaborative filtering, and knowledge-based strategies. Since traveling is often a group activity, families and groups of friends can receive group recommendations based on their combined profiles. The recommender system is tested in a prototype web application and evaluated by a group of test users. The results prove the usefulness of recommendations for travel destinations and show that the hybrid system outperforms each individual technique.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information Filtering; H.4 [**Information Systems Applications**]: Miscellaneous

Keywords

Recommender system, Hybrid, Travel, Tourism, Group recommendations

1. INTRODUCTION

Increasing amounts of information on traveling are available on the world wide web. As is the case for many other domains, the web is becoming the most important information source for planning a holiday. Specialized web sites, such as Expedia or SkyScanner, exist for finding the best deals, flight tickets or travel packages. Others, such as WikiVoyage or Frommers, are specialized in providing information and travel advice on different destinations. Reviews and evaluations of hotels, restaurants, and attractions can be read on websites such as TripAdvisor.

Although these services are all valuable information sources, they typically give no personal advice which holiday destination to chose. Here, recommender systems and artificial intelligence techniques [3] can help to overcome the problem of information overload and provide users valuable recommendations for destinations tailored to their personal preferences, requirements, and constraints.

Most research on recommender systems focuses on domains like movies, songs, or e-commerce. Specific characteristics of the domain make recommendations for travel destinations a lot harder. Firstly, data regarding travel destinations (metadata and ratings) are harder to acquire than the freely available dataset for movies such as MovieLens. Secondly, since most people travel only occasionally, the rating matrix is typically very sparse. Thirdly, users often have specific constraints (e.g., budget, distance) in addition to their personal preferences. And finally, traveling is typically a group activity: people often travel together. So group recommendations, combining the preferences of all group members, might be more suitable than individual recommendations.

The remainder of this paper is structured as follows. Section 2 gives an overview of related work. Section 3 provides an overview of the architecture of the travel recommender system and its internal data flow. Section 4 gives details about the data that is used and the data origins. In Section 5, the system is presented from the user point of view, with a focus on the features and the interface. The various recommendation algorithms are discussed in Section 6. Section 7 explains how the extension to group recommendations is realized. Section 8 gives the results of a user evaluation of the recommender systems. Finally, Section 9 draws conclusions from our research.

2. RELATED WORK

Various (group) recommender systems for points-of-interest (POIs), such as tourist attractions, restaurants, and hotels, have been proposed in literature. The Pocket Restaurant Finder provides restaurant recommendations for groups that are planning to go out eating together. The application can use the physical location of the kiosk or mobile device on which it is running, thereby taking into account the position of the people on top of their culinary preferences. Users have to specify their preferences regarding the cuisine type, restaurant amenities, price category, and ranges of travel time from their current location on a 5-point rating scale. When a group of people is gathered together, the Pocket Restaurant Finder pools these preferences together and presents a list of potential restaurants, sorted in order of expected desirability for the group using a content-based algorithm [14].

Intrigue is a group recommender system for tourist places which considers the characteristics of subgroups such as children or disabled and addresses the possibly conflicting preferences within the group. In this system, the preferences of these heterogeneous subgroups of people are managed and combined by using a group model in order to identify solutions satisfactory for the group as a whole [1].

Also in the context of tourist activities, the Travel Decision Forum is an interactive system that assists in the decision process of a group of users planning to take a vacation together [10]. The mediator of this system directs the interactions between the users thereby helping the members of the group to agree on a single set of criteria that are to be applied in the making of a decision. This recommender takes into account people's preferences regarding various characteristics such as the facilities that are available in the hotel room, the sightseeing attractions in the surrounding area, etc [9].

An alternative recommender system for planning a vacation is CATS (Collaborative Advisory Travel System) [15]. It allows a group of users to simultaneously collaborate on choosing a skiing holiday package which satisfies the group as a whole. This system has been developed around the DiamondTouch interactive tabletop, which makes it possible to develop a group recommender that can be physically shared between up to four users. Recommendations are based on the group profile, which is a combination of individual personal preferences.

The last example in the domain of POIs is Group Modeller, a group recommender that provides information about museums and exhibits for small groups of people [11]. This recommender system creates group models from a set of individual user models.

In contrast to existing systems, the goal of our recommender system, called TravelWithFriends, is to offer a more complete service delivering personalized recommendations for destinations taking into account the personal preferences, constraints, and feedback of the user. For each destination, travel distance, budget, and geographical location are considered and the local attractions and POIs are processed. Because of these domain specific characteristics, different recommender approaches are combined into a hybrid recommender. A group recommendation strategy is used to aggregate the preferences of different people who intend to travel together.

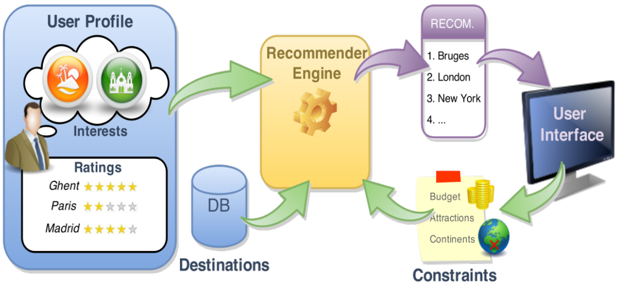


Figure 1: Overview of the data flow in the recommender system.

3. SYSTEM ARCHITECTURE AND DATA FLOW

Figure 1 shows the high-level flow of information through the recommender system. The recommender systems is fed with ratings and travel destinations coupled with metadata. Users interact with the system through the user interface. Personal constraints can be specified as input together with ratings for destinations. Recommendations are delivered as the output to the user, who can further give feedback on these recommendations.

Figure 2 zooms in on the recommender engine and the information flow within the recommender (red labels). The following subsequent steps can be indentified in the information flow.

1. Creating the user query: the user selects personal interests and destination constraints.

2. Constraint pre-filtering: the destinations in the database are checked against the constraints and a candidates shortlist is constructed.

3. Rating prediction: different recommendation algorithms calculate a rating prediction for the destinations of the shortlist.

4. Score merging: the rating predictions of the different algorithms are merged into one hybrid rating prediction.

5. Delivering recommendations: the destinations with the highest hybrid rating prediction are presented to the user as the final recommendations.

4. DATA STRUCTURE

The items, processed and output by the recommender, are all cities known for their tourism value. Many online services for POIs are available such as Google Places, Yelp, or Yahoo Local. Although these services contain lots of useful data, specific tourist information is often missing, such as information about tourist attractions in a city or the suitability of a location as a holiday destination.

As information source for our recommender service, we used the freely available data set of WikiVoyage [19], the Wikipedia alternative for travel destinations, which is available under an open license by the Wikimedia Foundation. This information service consists of more than 26,000 locations and tourist information pages, created by users. One

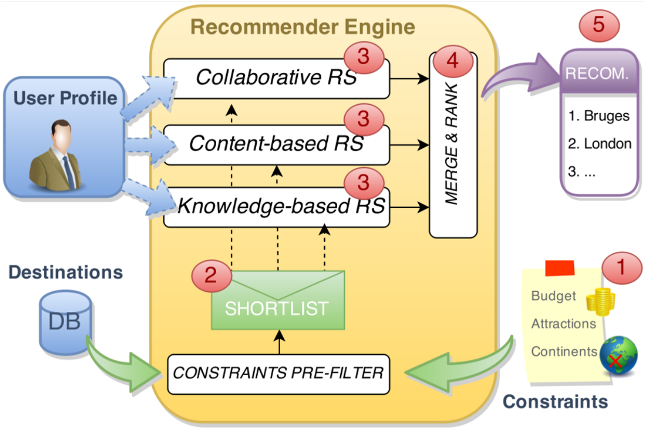


Figure 2: The system architecture of the recommender engine.

of the main advantages of this service, is that all entries have specific tourism value and come with information that is useful for tourists. However, many of these pages are not actual destinations but rather collections of destinations, information on a specific tour, etc. Therefore, a first filtering of the entries of WikiVoyage was performed using the database of GeoNames [7] in order to select only the actual destinations. GeoNames is a database listing over 100,000 place names in the world with their geographic data. The result of this first filtering was a set of 6,900 cities, towns, and villages.

Many of the resulting listings are minor, little-known locations, which may be interesting to explore while in the vicinity, but that have insufficient tourism value to be a travel destination on itself. Since these minor locations would be unsuitable as a recommendation for a travel destination, a second filter was necessary in order to only recommend 'sufficiently relevant' places. This filter used the popularity (measured by the number of ratings) on the popular website TripAdvisor [18], an American travel website providing reviews of travel-related content. The threshold for being considered as sufficiently relevant for a tourist destination was set to having at least 25,000 reviews on TripAdvisor. The resulting database contains 685 famous (and less famous) tourist locations, but can be easily extended with additional destinations (by relaxing one of the filters for example).

Regarding the information about the travel destinations, two crucial information resources are consulted:

- The *Travel Destination* database consists of general information about the destination, such as a description and location coordinates, as well as background information on the region and country.

- The *Domain Knowledge* database consists of specific domain knowledge such as a mapping of locations and typical tourist profiles, attraction types, and typical transport costs.

In order to obtain a typical tourist profile for each destination, the website Gogobot [8] is consulted. Gogobot is a travel application website that lets users rate travel destinations and attractions. In comparison with other social travel networks such as TripAdvisor, Gogobot differentiates by making use of *tribes*. Gogobot's 19 tribes represent tourist profiles (e.g., backpackers, family travelers, adventure travelers, business travelers, or budget travelers) to which users may relate. The tribe-specific information for a destination is obtained in two ways. On the one hand, users on Gogobot can explicitly specify that a destination is 'recommended for' a specific tribe such as Backpackers. On the other hand, Gogobot users can indicate in their own profile which tribes best match their interests. Destinations that received a positive rating from the user may also be suitable to other users who belong to (some of) the same tribes. In other words, we assume an implicit coupling between the user's tribes and the destinations that the user has rated. By gathering the tribe information of all users who rated the destination, a more detailed profile of the destination can be obtained. When combining tribe information of different users, the explicit tribe association was given twice the weight of the implicit association. In case of a user-item pair for which an explicit tribe recommendation as well as an implicit tribe association based on a star rating is available, only the explicit tribe recommendation is used.

For travel costs, a specialized information service was used. Various web services provide real-time prices for trains, airplanes, or another means of transportation. The webservice Rome2rio [17], which was used in TravelWithFriends, combines different transportation methods and predicts the travel cost between any two locations in the world. It taps into the information of many different online services and databases to gather information on flights, trains, buses, boats, and even taxi fares to come up with all possible means to reach your destination.

The users, who interact with the system and receive recommendations, are also represented by two information resources:

- The *User Rating* database keeps track of the 5-star ratings of all users given to travel destinations, as well as implicit feedback that indicates which places the user has visited (without star rating).

- The *User Profile* database stores more general information about each user such as login information, explicitly stated interests, and demographic data.

To reduce cold-start difficulties of our recommender system, ratings and implicit feedback (selecting "Been here" to indicate that you have visited the location) from Gogobot were used. More than 300,000 ratings by 1759 users from Gogobot were imported. Ratings for attractions were aggregated to ratings for the destination were the attraction can be found. Ratings for destinations that are not in the destination database (or filtered out because of their low tourism value), are redundant and ignored in the calculations. Finally, 53,028 ratings were imported into the recommender system.

5. TRAVELWITHFRIENDS WEB APPLICATION

The TravelWithFriends recommender system is made available for end-users through a web application accessible in a standard web browser. The web application consists of many pages, such as the register page, a page for creating and joining groups, and the traditional search functionality.

In comparison with other recommender domains, traveling is for many users a less frequent activity compared to

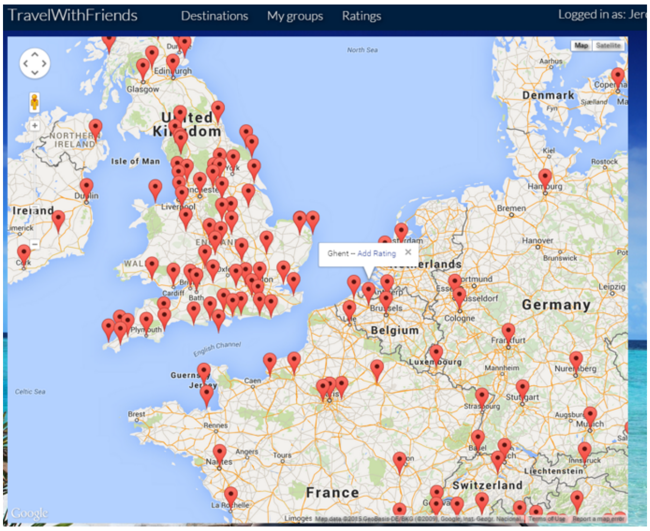
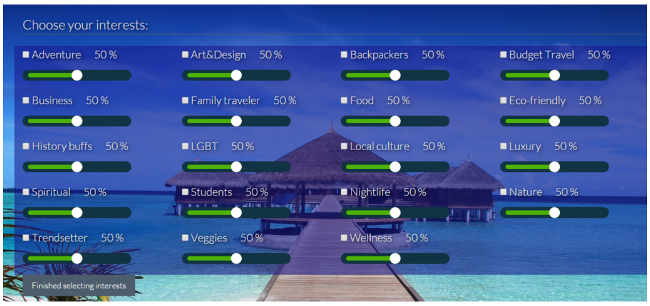


Figure 3: Screenshot of the Google Maps based user interface for destination selection.



Choose your interests:

- Adventure 50 %
- Art&Design 50 %
- Backpackers 50 %
- Budget Travel 50 %
- Business 50 %
- Family traveler 50 %
- Food 50 %
- Eco-friendly 50 %
- History buffs 50 %
- LGBT 50 %
- Local culture 50 %
- Luxury 50 %
- Spiritual 50 %
- Students 50 %
- Nightlife 50 %
- Nature 50 %
- Trendsetter 50 %
- Veggies 50 %
- Wellness 50 %

Figure 4: Screenshot of the user interface showing the possibility to explicitly specify preferences.

listening to songs or watching movies, thereby exacerbating the sparsity problem. To reduce the sparsity, users can state their previous travel experiences by giving ratings to destinations they have visited in the past. Users can search for these locations by name, or alternatively, they can navigate to the location through Google Maps, as demonstrated in Figure 3.

To bootstrap the content-based recommender component, users can also indicate their interests for 19 travel categories, which are used as an initial profile. Figure 4 shows a screenshot of the user interface illustrating the explicit profile preferences of the user. For each of the typical travel interests, users can specify their affinity.

In addition, users can specify personal constraints regarding the travel destination, such as budget, the continent of the destination, and the presence of specific kinds of attractions.

6. RECOMMENDER ENGINE

To cope with the complex aspects of travel recommendations, such as the desired serendipity, the sparsity problem, and user constraints, multiple recommender approaches are combined into a weighted hybrid recommender. Collaborative filtering can introduce serendipity into the recommendations by comparing consumption data of similar users. The content-based approach can better handle the sparse data matrix. User constraints are taken into account by the knowledge-based recommender (and a pre-filter).

In a production environment, users will receive only one list of (hybrid) recommendations. However, for evaluation purposes, the test users (cfr. Section 8) received five recommendations lists so that each algorithm could be evaluated separately: collaborative filtering, content-based, knowledge-based, hybrid, and a static list of the most popular travel destinations. Users can select a destination from the recommendation list to request more information, or can give feedback on the recommendations.

Before applying the recommendation algorithms, a pre-filtering of the candidate destinations is performed. The system eliminates destinations that the user has already visited (assuming these would make for undesirable recommendations) as well as destinations that do not fulfill the user's constraints. In the next phase, the various recommendation algorithms will restrict their selection to destinations that made the shortlist.

6.1 Collaborative filtering

For collaborating filtering, we opted for an item-item approach and used the implementation of the Lenskit Framework [6]. As input, a combination of explicit ratings and implicit feedback is used, given by users on Gogobot.com [8]. For the explicit ratings, the system uses ratings for the destination (the city), as well as ratings for the attractions at the destination which are averaged into one rating. In comparison with other domains, such as online shop items or music, the ratings gathered from Gogobot are much more positive (more than 90% is ≥ 3).

In addition, for each destination or attraction at a destination, users can indicate on Gogobot if they have been there. This data is used as implicit feedback on the destination in TravelWithFriends, because it contains two pieces of information: 1. the user has been to this place, and recommending this destination again is therefore undesirable and 2. the user has shown interest in this place by visiting it.

The item-based collaborative approach takes two phases to predict the rating of a user for a given destination. In the first phase, a collection of most-similar destinations, called the k-nearest neighbors, is determined. These neighbors are selected by calculating a similarity measure between each pair of destinations and selecting the ones with the highest similarity value. For the neighbor selection, explicit ratings combined with implicit feedback were used in order to reduce the sparsity of the matrix. In our approach, explicit and implicit feedback are mapped to a binary value: 1 if the destination was rated or tagged as "Been here"; 0 otherwise. Since ratings are mainly positive, the mapping of 5-star ratings to a binary value is not considered as a loss of information for calculating item similarities. Because of these binary values, the traditional Pearson correlation is not feasible. Because some items (popular destinations) received much more feedback than others, the cosine similarity is not the optimal correlation measure to ensure that less popular destinations have a fair chance to get recommended. Another approach is to use the conditional probability $P(j|i) = \frac{P(i,j)}{P(i)}$ to calculate how likely a specific destination is, in case another destination has been visited. This function evaluates whether two items are associated, but still favors destinations with a large amount of feedback, i.e. highly popular destinations.

To correct for the popularity of the item j, a modified version of the conditional probability with an additional term $P(j)^{\alpha}$ was used as similarity measure:

$$sim(i,j) = \frac{P(i,j)}{P(i) * P(j)^{\alpha}} \tag{1}$$

In the implementation, $\alpha = 0.2$ provided the best balance between constraining the popularity and measuring similarity based on empirical research.

In the second phase, the rating prediction will be calculated based on these most similar destinations. In our implementation, $k = 20$ was chosen as this is a typical value for the k-nearest neighbors algorithm. We denote $\mathcal{N}_u^e(i)$ as the neighborhood of destination i for user u. This neighborhood consists of the items j that the target user u has explicitly rated, and that are most similar to the item i. As such, this neighborhood is different for each user. Next, the weighted sum scoring function with mean centering [5] is used for items that received an explicit rating from the target user in order to make a rating prediction $\hat{r}^e$.

$$\hat{r}^e_{u,i} = \overline{r}_i + \frac{\sum\limits_{j \in \mathcal{N}_u^e(i)} sim(i,j)(r_{u,j} - \overline{r}_j)}{\sum\limits_{j \in \mathcal{N}_u^e(i)} |sim(i,j)|} \tag{2}$$

To also take into account the implicit feedback, a second scoring function was used for the binary data.

$$\hat{r}^i_{u,i} = \frac{\sum\limits_{j \in \mathcal{N}_u^i(i)} sim(i,j) * \overline{r}_j}{\sum\limits_{j \in \mathcal{N}_u^i(i)} |sim(i,j)|} \tag{3}$$

Here, the neighborhood $\mathcal{N}_u^i(i)$ stands for the items j for which the target user has provided implicit feedback or an explicit rating, and that are most similar to the item i. Notice that the user's neighborhood for implicit feedback $\mathcal{N}_u^i(i)$ can be different from the user's neighborhood for explicit feedback $\mathcal{N}_u^e(i)$, since a user might have provided implicit feedback for different items compared to the user's ratings.

Finally, the weighted sum of both rating predictions is calculated to combine explicit ratings and implicit feedback, as is commonly done [20].

$$\hat{r}_{u,i} = \frac{\alpha * \hat{r}^e_{u,i} + \beta * \hat{r}^i_{u,i}}{\alpha + \beta} \tag{4}$$

The weights α and β were set to: $\alpha = 2 * \#\mathcal{N}_u^e(i)$ and $\beta = \#\mathcal{N}_u^i(i)$. The values of α and β were chosen so that if both neighborhoods contain the same number of items (i.e. 20 if enough neighbors can be found), then the rating prediction for the explicit ratings contributes for 2/3 versus only 1/3 for the prediction based on implicit feedback. If however, the neighborhood for the explicit ratings has far fewer similar items than the one for implicit feedback, then the weight is shifted more towards the rating prediction with the implicit feedback.

If data sparsity prevents finding an extensive neighborhood, and $\mathcal{N}_u^i(i)$ contains fewer than 5 similar items (which implies $\mathcal{N}_u^e(i)$ has also less than 5 items), then the collaborative filtering approach is considered unreliable. In this case, recommendations using collaborative filtering might not be accurate enough given the small neighborhood size and the recommendations are disregarded. The recommender system will then fall back on the content-based and knowledge-based approaches.

6.2 Content-based recommender

The idea of content-based recommendation approaches is to find matches between features of a particular item and the user's profile. If item features are not directly available, they are often obtained by analyzing textual descriptions of the items and extracting keywords from them. This approach can also be applied in the domain of travel destinations, but has been shown to deliver often irrelevant or overly obvious features. Therefore, in TravelWithFriends another approach, specially tailored for the domain of travel destinations, was adopted.

The approach is based on the idea to characterize a travel destination by the categories and keywords linked to the POIs at the destination. These POIs are often accurately annotated by specialized information services and are often the main incentive to visit a travel destination. TravelWithFriends utilizes the tags of attractions described on TripAdvisor [18], but similar information sources can be a valuable alternative. The tags of attractions on TripAdvisor are chosen from a fixed set of attraction categories and are restricted to one tourism topic. We illustrate our approach using Paris as a potential destination. Among its most prominent tourist attractions are the world famed museums 'musée du Louvre' (categorized as [Art Museum, Museums] on TripAdvisor), and 'musée d'Orsay' [Speciality Museum, Museums]. Paris features also some well known landmarks such as the 'Eiffel tower' [Points of Interest & Landmarks, Sights & Landmarks], 'Arc de triomphe' [Architectural Buildings, Historic Sites, Sights & Landmarks] and the 'Notre Dame Cathedral' [Religious Sites, Sights & Landmarks]. These key attractions and their associated tags already give a good overview of what Paris has to offer to tourists.

The relative importance of a tag for an item is typically determined by a measure such as the TF-IDF (Term Frequency - Inverse Document Frequency) [12]. To increase the contribution of the more famous and popular attractions at the destination, the tag frequency is multiplied by the number of reviews for the coupled attraction. In the example of Paris, the tag 'Speciality Museums' (attached to musée d'Orsay) was applied 26,149 times (the number of reviews for musée d'Orsay) to Paris. In contrast, the tags applied to the Parc des Buttes Chaumont (i.e. the 50th most popular attraction in Paris) only receives a weight of 548, the number of reviews for Parc des Buttes Chaumont.

Because of a large variation in the frequency of occurrence of tags and reviews, a minor change was made to the traditional TF-IDF by taking the square root of the frequency term, $f_{t,d}$ to reduce the influence of the absolute review frequency. This results in the following formula for the TF-IDF weight for tag t of destination d, part of the collection of all destinations D. Here, N is the number of destinations in D, and $f_{t,d}$ is the frequency of tag t in destination d, which means the frequency of tag t in all attraction descriptions of destination d, multiplied by the number of reviews for that attraction.

$$TFIDF(t,d,D) = \sqrt{f_{t,d}} * log_2 \frac{N}{|\{d \in D : t \in d\}|} \tag{5}$$

The necessity to take the square root of the term frequency can be illustrated by an example. If the tag frequency, multiplied by the number of reviews, is used in combination with the traditional TF-IDF, then the weight of a few top attractions is too high, thereby neglecting the contribution of other attractions at the destination. If any of these top attractions has a rare tag (and thus a very high IDF), this tag will dominate the recommendations. For 'Barcelona' as destination, for instance, the 'Sagrada Familia' is one of the top attractions, which has a rather rare tag 'Religious Sites'. This tag will dominate the recommendations in case of the traditional TF-IDF, leading to "similar" destinations, all renown for their beautiful cathedrals including 'Santiago de Compostela', 'Cologne', and 'Rouen'. Since Barcelona offers much more than the 'Sagrada Familia', this biased reflection was undesirable.

The logarithm of the term frequency has been proposed as an alternative weight for the term frequency in literature [12]. However, experiments showed that the logarithm shifted too much weight to less popular attractions. Analysis of the resulting recommendations showed that for the domain of travel destinations, the square root of the term frequency provides the right balance between both popular and less popular attractions. The square root reduces the weight of top attractions, but preserves a sufficiently large difference in contribution compared to less significant attractions.

In the same manner, the derived destination tags are used to build a content-based user profile based on the destinations that are positively rated (≥ 3.5) by the user. For all these positively rated destinations, the TF-IDF values are summed per tag in the user profile. Finally, the derived destination tags are compared with the user profile using the traditional cosine similarity. The resulting similarity score is transformed to the range $[1-5]$ and used as content-based rating prediction.

6.3 Knowledge-based recommender

The knowledge-based recommenders makes use of deeper connections and information provided by domain experts. Just like a human travel agent typically asks customers for their target budget, travel distance and accommodation expectations, the knowledge-based recommender will select destinations in a similar matter. This user input can be defined as a hard constraint or as a soft constraint (rather a guideline for the recommender). The pre-filter eliminates all destinations that do not fulfill the hard constraints before the candidate destinations are handed over to the recommendation algorithms. The soft constraints are handled by the knowledge-based recommender, which gives a penalty to destinations that are a good match for the user's preferences, but do not completely fulfill the requirements. So, destinations for which the soft constraints are not met, can still end up in the final recommendation list if they match the user's preferences.

In comparison with collaborative filtering and the content-based recommender, the knowledge-based recommender collects information specific for the domain of travel destinations, and is therefore not directly applicable to other domains. The following information sources were integrated into the knowledge based recommender:

1. Geographic information: the exact location (longitude and latitude), continent, and country of each destination.

2. Travel costs: the costs of traveling from your current location to the destination in question.

3. Attraction types: what specific attraction types can be found at that destination.

4. Tourist profile (stereotypes such as Backpackers, Family Travelers, etc.): to what degree each location matches typical tourist profiles as defined in Gogobot [8].

Constraints regarding the location and distance, as well as the traveling cost can be specified by the user in the interface of the application, as showed in Figure 5. Requirements regarding the types of attractions available at the destination, such as beaches, amusement parks, etc., can be selected using check-boxes. These constraints and user requirements are matched against the candidate destinations, providing a score for each dimension (location, costs, profile, attractions).

Table 1 shows the scoring function for each dimension, as well as a weight for the relative contribution of each dimension to the rating prediction. For the location dimension, a score function is proposed that decreases as the travel distance exceeds the *max_distance* as defined by the user. The square root allows destinations that are only slightly further than the *max_distance*, by assigning only a small penalty to these. For the cost dimension, a score function is proposed that decreases linearly as soon as the expected cost exceeds the predefined budget of the user. For the attractions available at the destination, the scoring is the ratio of the number of attractions that are requested and available, and the total number of attractions that are requested by the user.

For the tourist profile, each user is linked to one or more typical profiles (e.g., 30% Backpackers, 70% Adventure Travelers). This mapping to typical profiles can be performed in two ways. Users have the option to manually select what profiles they believe best match their interests (Figure 4). Alternatively, the typical profiles can be selected automatically by matching the user's explicit ratings with the typical profiles of the rated destinations. This approach is similar to the profile creation based on tags, used in the content-based recommender.

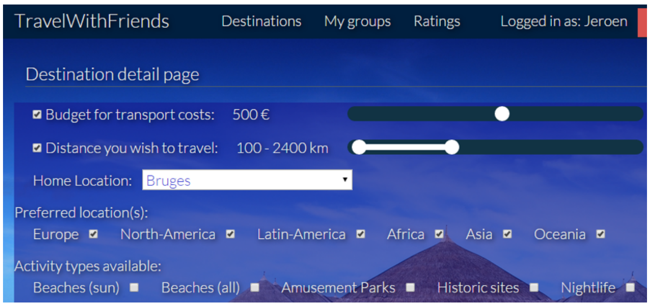
For each candidate destination a typical tourist profile is calculated based on the typical profiles of the users who rated the destination on Gogobot [8]. E.g., if 80% of the users who positively rated the destination are Backpackers, then it is classified as a 80% Backpackers destination. Subsequently, the user's typical profile is compared with the destination's typical profile using the cosine similarity. The scores of the different dimensions are combined using a weighted average to calculate the knowledge-based rating prediction.

$$\hat{r}_{u,i} = \frac{\sum\limits_{k \in \mathcal{D}} w_k * sc(i,k)}{\sum\limits_{k \in \mathcal{D}} w_k} \tag{6}$$

Here, the summation is limited to the dimensions D, for which constraints are specified by the user. The weights of the different dimensions are specified in Table 1. In our implementation, all weights have the same value, except for the

Table 1: The scoring function of the components of the knowledge-based recommender

Dimension	Scoring sc(i,k)	w_k
Geo location	$1 - \sqrt{\frac{max(distance(user, location) - max\_distance, 0)}{max\_distance}}$	1
Travel costs	$1 - \frac{max(expected\_cost - max\_budget, 0)}{max\_budget}$	1
Attractions	$\frac{\#typesmatched}{total\#typesrequested}$	$\frac{1}{2}$
Tourist profile	cosine sim(item, profile)	1



☑ Budget for transport costs: 500 €

☑ Distance you wish to travel: 100 - 2400 km

Home Location: Bruges

Preferred location(s):
Europe ☑ North-America ☑ Latin-America ☑ Africa ☑ Asia ☑ Oceania ☑

Activity types available:
Beaches (sun) □ Beaches (all) □ Amusement Parks □ Historic sites □ Nightlife □

Figure 5: Screenshot of the user interface showing the options to define user constraints.

weight of the attractions. Since users might specify multiple attraction types that are sometimes hard to combine (e.g., beaches, amusement parks, and historic sites), the weight of the attraction dimension was decreased to 1/2.

6.4 Hybrid recommender

While the three individual recommendation approaches each generate a rating prediction, merging their output combines the different information sources and should make up for misjudgements of the individual recommenders. To merge the rating predictions, a simple weighted sum of all three predicted scores is calculated. The different indices are cf for collaborative filtering, cbf for content-based filtering, and kb for the knowledge-based recommender.

$$\hat{r}_{hybrid} = w_{cf} * \hat{r}_{cf} + w_{cbf} * \hat{r}_{cbf} + w_{kb} * \hat{r}_{kb} \tag{7}$$

$$w_{cf} + w_{cbf} + w_{kb} = 1 \tag{8}$$

These weights are not static, but influenced by the available data. If enough data is available for all recommenders, each algorithm will contribute for 1/3 to the hybrid recommendations. If only a limited amount of neighbors are found for collaborative filtering, w_{cf} is lower, or even zero if less than 5 neighbors can be found. The knowledge-based recommender has a lower contribution ($w_{kb} < 1/3$) if fewer (soft) constraints are specified by the user. Since an initial profile is created for each user, the content-based recommender usually has sufficient information to generate recommendations and can therefore act as the fall-back algorithm when both other approaches show little confidence.

7. GROUP RECOMMENDATION

Many travel plans are not made by individuals but by groups of people: friends, families, sports teams, etc. Besides individual recommendations, TravelWithFriends therefore allows users to create groups or join existing groups of friends and receive recommendations for the whole group. For most people, choosing a travel destination is an important decision, in which communication among the group members is essential. Group members typically want to discuss the destination thereby communicating their concerns and preferences based on some concrete suggestions. Therefore, group recommendations are generated in two subsequent phases. First, the system makes a shortlist of destinations for the group, based on the recommendation lists of each individual group member. Second, the group recommender acts as a conversational recommender. Each group member has the opportunity to provide feedback and rank this shortlist of candidates, after which the system makes a fair and balanced review and presents the final recommendations.

The process of generating group recommendations is illustrated in Figure 6. In the first phase, recommendations of individual users are merged into group recommendations using a recommendation aggregation technique [4]. We opted to aggregate the individual recommendation lists into a group recommendation list instead of aggregating the individuals' ratings into group ratings and subsequently generating group recommendations from these group ratings [4]. The reason for this is that aggregated recommendations for a group can be linked to recommendations for individuals, and as a result, can be explained more clearly in terms of an individual's preferences and constraints.

To aggregate the individual recommendations into group recommendations, various strategies are possible [13]. However, some have obvious disadvantages. Using the average of each member's rating prediction as a rating prediction for the group, i.e. the 'average' strategy, has the disadvantage of individuals who might be very unhappy with the final choice. If one user has a strong aversion to a particular destination, but the other group members love it, then this destination might still be recommended because of a high average rating prediction. Leaving one of the group members really unhappy about the destination is an unwanted situation. Therefore, the 'average without misery' strategy is employed as recommendation aggregation method, since this strategy cares about fairness and avoiding individual misery [13]. This strategy calculates the average of each member's rating prediction, but eliminates the destination if one of the group members has a rating prediction below a threshold. The threshold was chosen at 50% of highest scoring destination for that user. This way, destinations that are strongly disliked by any of the group members are eliminated from the group recommendations. Based on the assumption that users want recommendations for destinations they have not yet visited, destinations that have already received feedback or a rating from one of the group members are also eliminated. The result is a list of ten candidate destinations which are offered as an initial recommendation list

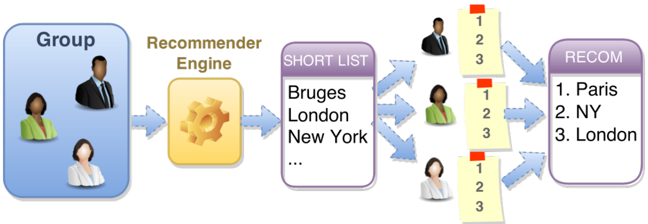
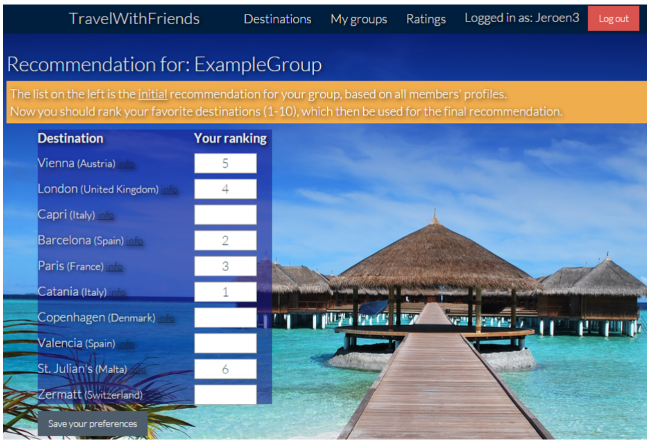


Figure 6: Schematic overview of the generation of group recommendations.



Recommendation for: ExampleGroup

The list on the left is the initial recommendation for your group, based on all members' profiles.
Now you should rank your favorite destinations (1-10), which then be used for the final recommendation.

Destination	Your ranking
Vienna (Austria) info	5
London (United Kingdom) info	4
Capri (Italy) info	
Barcelona (Spain) info	2
Paris (France) info	3
Catania (Italy) info	1
Copenhagen (Denmark) info	
Valencia (Spain) info	
St. Julian's (Malta) info	6
Zermatt (Switzerland)	

Figure 7: Screenshot of the user interface showing the possibility to rank the group recommendations.

to the group.

In the second phase, group members can give feedback on the list and indicate their favorites. In order to give users the opportunity to negotiate the travel options, each member is invited to give a personal ranking (from 1 to 10) to the candidate group recommendations, as shown in Figure 7. The users' ranking of the recommendations are processed by the Borda count election method. The Borda count method determines the winner(s) of the election by giving each candidate a number of points corresponding to the number of candidates ranked lower [13]. Based on the resulting Borda count, the group is finally presented their top-5 destinations to reduce the choice overload of the final recommendation list.

8. EVALUATION

The TravelWithFriends application was presented to 16 users who were asked to experiment with the recommender system and evaluate the different recommendation lists (3 algorithms, hybrid recommendations, and a static list of popular destinations). The evaluation was performed in two phases.

The goal of the first phase was to assess the general quality of the travel recommender system. To collect some qualitative feedback regarding the service, each user was asked to fill in a questionnaire, based on the evaluation framework of Pu [16]. Figure 8 shows the results of four multiple-choice questions assessing the general quality of the system (not about a specific recommendation algorithm). All users were overall satisfied with the system (Figure 8(a)). Their comments (not shown here) were positive about the possibility to explore new, unfamiliar destinations. They enjoyed the experience of determining their next travel destination using the service.

Next, the results show that most users consider it easy enough to specify their preferences. However, there is some room for improvement here (Figure 8(b)). The open questions indicated users would like more options for choosing their type of holiday (citytrip vs. hiking trip), the option of a general safety advice, the tourist-friendliness of the destination, and the option to determine the duration of the trip.

In addition, most users are convinced that the recommendations are useful and a suitable candidate for their travel destination (Figure 8(c)). Adding explanations to the recommendations can be an improvement to further increase the users' trust in the system. Finally, almost all users also indicated they would use the application if it became publicly available (Figure 8(d)).

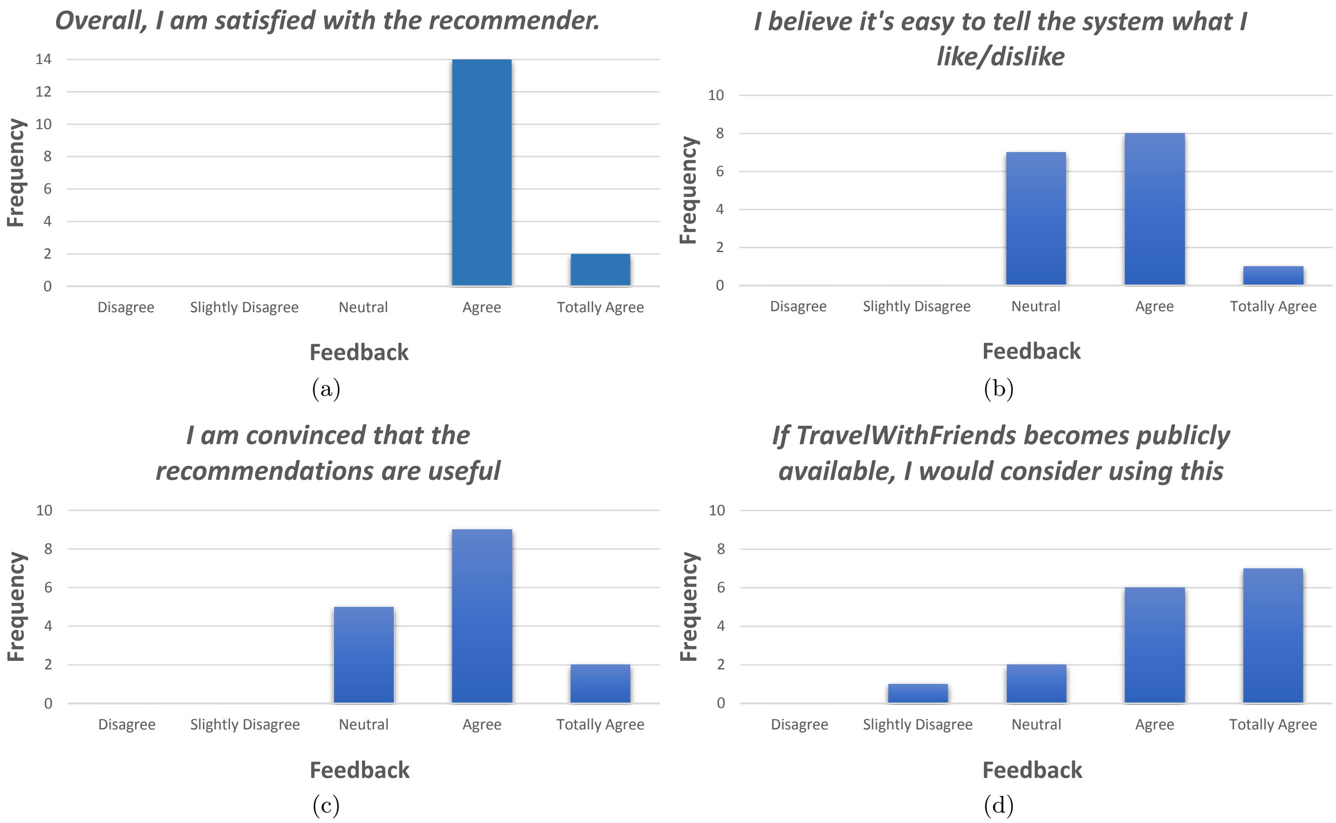
The goal of the second phase was to assess the users' opinion about the quality of each recommendation algorithm: collaborative filtering (CF), content-based filtering (CBF), the knowledge-based recommender (KB) and the hybrid combination of these algorithms (HYB). As a baseline to compare the different algorithms, a fifth approach was included, which simply returned the static, non-personalized top of most-popular destinations (TOP). This list shows the most rated destinations on TripAdvisor, excluding the destinations already rated by the user.

Users were invited to use the application, starting with the preparatory steps of adding some ratings, selecting interests, and specifying constraints. Subsequently, users could explore the recommendations generated based on their input. To compare the different recommendation algorithms, users in this test were presented with five different lists of eight recommendations each. Eight recommendations is considered as an optimal number to prevent choice overload, while providing users different options and the coupled choice satisfaction [2]. These five lists were randomly shuffled and presented without any hint of the algorithm that was used to produce the list in order to obtain unbiased evaluation results.

The test users were asked to rank these five lists based on their own assessment of the most suitable recommendations. Figure 9 shows the distribution of the obtained rankings for each algorithm. These results indicate that the hybrid algorithm is most appreciated by the test users with 6 users choosing this as the best option, and 5 more users rewarding this algorithm with a second place.

Besides the hybrid recommender, also the content-based and knowledge-based recommender were liked by many users, whereas the TOP approach achieved the worst results (as expected). We hypothesize that content-based and knowledge-based recommendations score better than the collaborative filter because users recognize their constraints and personal preferences in these recommendations.

A statistical analysis using the Student's t-test was performed to test the superiority of the recommendation algorithms against the baseline approach (TOP). The mean of the rankings assigned by the users was compared for the different algorithms. The null-hypothesis was that the differences in mean ranking were merely due to randomness of



(a) (b) (c) (d)

Figure 8: The results of the user evaluation regarding the general quality of the travel recommender.

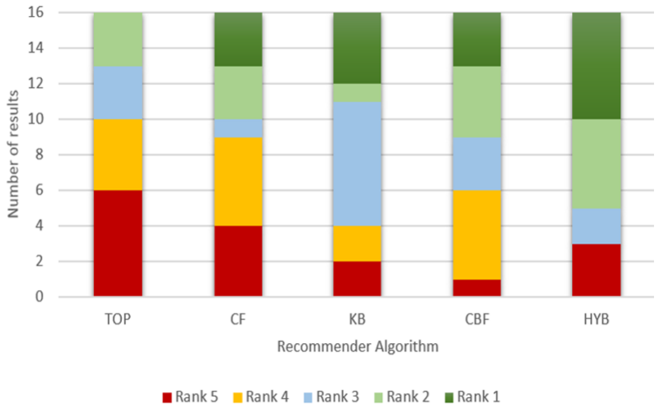


Figure 9: Distribution of the rankings given to the recommendation algorithms by the test users. Rank 1 is the best; Rank 5 is the worst.

the results. The t-tests showed that the difference with the baseline recommender (TOP) in terms of mean ranking was statistically significant for the hybrid recommender (p-value = 0.004), the content-based recommender (p-value = 0.028), and the knowledge-based approach (p-value = 0.031). Only the collaborative filter did not show statistical evidence (p-value = 0.251) of receiving a better ranking than the baseline.

9. CONCLUSIONS

Because travel destinations proved to be a complex domain for recommendations, characterized by personal preferences, user constraints, and being a group activity, no single algorithm would be able to consider all aspects. Moreover, gathering metadata and user feedback (ratings) showed to be less trivial for travel destinations than for more classical recommender domains such as movies or books. A hybrid system, combining different recommender approaches supplemented with the ability to generate group recommendations, was proposed.

User testing showed the usefulness of the proposed travel recommender system. Users enjoyed the new approach for discovering destinations and were happy to explore new places to consider as a travel destination. A comparison of different recommendation algorithms indicated that users prefer the hybrid recommendations above content-based, knowledge-based, and collaborative filtering recommendations. Differences in recommendation quality between these algorithms and an unpersonalized list of the most-popular destinations are clearly noticeable to the users. User comments argued for the inclusion of explanations of the recommendations in future versions of the application. Another option for future work is to recommend close-by locations, a multi-destination holiday, or a wider region to explore. In addition to the evaluation by individual users, the system will be tested by groups of users to evaluated the two phase group recommendation process in the future. Given the impact of the knowledge-based approach, it will be considered for pre-filtering plus weight initialization of the destination candidate set, rather than a recommender itself. Also the combination of both knowledge-based and content-based techniques will be investigated, because collaborative filtering seems to decrease the satisfaction of the users. Finally, we plan a performance evaluation to make the system useful as an actual product.

10. ACKNOWLEDGMENTS

We would like to thank Jeroen Dhondt for the work he performed in the context of this research during his master thesis.

11. REFERENCES

[1] L. Ardissono, A. Goy, G. Petrone, M. Segnan, and P. Torasso. Tailoring the recommendation of tourist information to heterogeneous user groups. In S. Reich, M. Tzagarakis, and P. De Bra, editors, *Hypermedia: Openness, Structural Awareness, and Adaptivity*, volume 2266 of *Lecture Notes in Computer Science*, pages 228–231. Springer Berlin Heidelberg, 2002.

[2] D. Bollen, B. P. Knijnenburg, M. C. Willemsen, and M. Graus. Understanding choice overload in recommender systems. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys '10, pages 63–70, New York, NY, USA, 2010. ACM.

[3] J. Borràs, A. Moreno, and A. Valls. Intelligent tourism recommender systems: A survey. *Expert Systems with Applications*, 41(16):7370–7389, 2014.

[4] T. De Pessemier, S. Dooms, and L. Martens. Comparison of group recommendation algorithms. *Multimedia Tools and Applications*, 72(3):2497–2541, 2014.

[5] C. Desrosiers and G. Karypis. A comprehensive survey of neighborhood-based recommendation methods. In F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors, *Recommender Systems Handbook*, pages 107–144. Springer US, 2011.

[6] M. D. Ekstrand, M. Ludwig, J. Kolb, and J. T. Riedl. Lenskit: A modular recommender framework. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys '11, pages 349–350, New York, NY, USA, 2011. ACM.

[7] GeoNames. The geonames geographical database, 2015. Available at `http://www.geonames.org/`.

[8] Gogobot Inc. The go-to place for places to go, 2015. Available at `http://Gogobot.com/`.

[9] A. Jameson. More than the sum of its members: challenges for group recommender systems. In *Proceedings of the working conference on Advanced visual interfaces*, AVI '04, pages 48–54, New York, NY, USA, 2004. ACM.

[10] A. Jameson, S. Baldes, and T. Kleinbauer. Two methods for enhancing mutual awareness in a group recommender system. In *Proceedings of the working conference on Advanced visual interfaces*, AVI '04, pages 447–449, New York, NY, USA, 2004. ACM.

[11] J. Kay and W. Niu. Adapting information delivery to groups of people. In *Proceedings of the Workshop on New Technologies for Personalized Information Access at the Tenth International Conference on User Modeling*, 2006.

[12] C. D. Manning, P. Raghavan, H. Schütze, et al. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

[13] J. Masthoff. Group modeling: Selecting a sequence of television items to suit a group of viewers. In *Personalized Digital Television*, volume 6 of *Human-Computer Interaction Series*, pages 93–141. Springer Netherlands, 2004.

[14] J. McCarthy. Pocket restaurantfinder: A situated recommender system for groups. In *Proceedings of the Workshop on Mobile AdHoc Communication at the 2002 ACM Conference on Human Factors in Computer Systems*. ACM, 2002.

[15] K. McCarthy, M. Salamo, L. Coyle, L. McGinty, B. Smyth, and P. Nixon. Cats: A synchronous approach to collaborative group recommendation. In G. Sutcliffe and R. Goebel, editors, *FLAIRS Conference*, pages 86–91. AAAI Press, 2006.

[16] P. Pu, L. Chen, and R. Hu. A user-centric evaluation framework for recommender systems. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys '11, pages 157–164, New York, NY, USA, 2011. ACM.

[17] Rome2rio. Rome2rio api and whitelabel, 2015. Available at `http://www.rome2rio.com/documentation`.

[18] TripAdvisor Inc. Tripadvisor website, 2015. Available at `http://www.tripadvisor.com/`.

[19] Wikivoyage Inc. Wikivoyage the free worldwide travel guide that anyone can edit, 2015. Available at `http://dumps.wikimedia.org/enwikivoyage/`.

[20] M. Zanker and M. Jessenitschnig. Collaborative feature-combination recommender exploiting explicit and implicit user feedback. In *Commerce and Enterprise Computing, 2009. CEC '09. IEEE Conference on*, pages 49–56, July 2009.