DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Data-Driven Destination Characterization in a Conversational Recommender System

Saadi Myftija

DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Data-Driven Destination Characterization in a Conversational Recommender System

Datengetriebene Charakterisierung von Reisezielen in einem konversationellen Empfehlungsdienst

Author:	Saadi Myftija
Supervisor:	Prof. Dr.-Ing. Jörg Ott
Advisor:	M.Sc. Linus Dietz
Submission Date:	15.01.2019

I confirm that this master's thesis in informatics is my own work and I have documented all sources and material used.

Munich, 15.01.2019 Saadi Myftija

Acknowledgments

I would like to express my sincere gratitude to my thesis advisor, Linus Dietz, for his valuable and constructive suggestions during the writing of this thesis.

I would also like to thank my family for the everlasting support they have given me throughout these years.

Abstract

In today's age of information, recommender systems have evolved from a nice-to-have feature to an essential necessity of many platforms. They are becoming increasingly ubiquitous and span through a myriad of domains, with e-commerce and tourism being two prominent ones. Travel recommender systems — given the nature of the domain — typically generate destination recommendations based on information about the items to be recommended. A lot of systems rely on domain experts to infer this information. This enables achieving a high quality of recommendation, but a common drawback is that it hinders the scalability of the systems.

In this thesis, we study possible approaches that could be used to reduce the reliance of travel recommender systems on expert knowledge. Specifically, we propose a data-driven approach for characterizing travel destinations based on inferring information from single points of interest. We acquire data for more than nine million Foursquare venues across approximately 200 cities worldwide, and integrate data sources for cost and climate information to come up with a characterization of the cities. We apply cluster analysis on the resulting dataset and interpret the five emerging city clusters.

Furthermore, we demonstrate how the dataset generated from this characterization process can be used for generating recommendations by building a prototype recommender system for cities. We use this prototype to conduct a user study to evaluate the performance of the recommender system and investigate the effect that critiquing has on the perceived quality of recommendations. The assessment criteria is based on the ResQue evaluation framework for recommender systems. Based on a statistical analysis of the data gathered from 76 participants of the user study, we were able to demonstrate that the recommender system which employs critiquing in its interaction flow achieves a significantly higher perceived accuracy of recommendations ($p = 0.01547$) compared to the baseline system that uses a traditional method of interaction.

Contents

1. Introduction

Traveling is a highly rewarding experience for many people. It is a fascinating gateway to exploring new places and cultures, taking a break from daily routines of life, pursuing self-fulfillment, and as such, it plays an important role in defining one's character. In their travels, philosophers and writers have found the inspiration for their ideas that advocate peace, coexistence and reciprocal respect. Not only does traveling help to bring people together, but it also encourages celebrating the differences between each other. Since ancient times, it has served as a bridge between nations, a way to exchange goods and knowledge, as well as a medium to share ideas about humanity. Today, travel and tourism continue to shape the economies of the world by accounting for more than 10% of the global GDP [62].

However, traveling is also an activity which requires a lot of planning beforehand. Choosing a destination can be a daunting task for tourists, as they have to filter through an exhausting amount of information, find options that match their preferences and balance potential cost or time constraints, all while dealing with the uncertainty associated to their information sources. While the abundance of tourism data in the internet creates more information sources to choose from, it also causes a severe information overload problem, which contributes towards making the planning task increasingly complex [7].

To tackle these challenges, recommender systems have become popular in the tourism domain as a solution to help tourists evaluate the overwhelming amount of information and choices they are presented with [11]. These systems utilize personalization techniques to create an understanding of tourists' preferences and enable them to make informed decisions when planning their travels. Typically, travel recommender systems focus on assisting the user on a specific step of the planning, e.g., choosing a destination, creating a route plan, or finding POIs (points of interest) when the tourist is already at the destination.

Many of the existing travel recommender systems consider the destination already as given, and focus on recommending POIs instead [11]. There is distinguishable absence of systems recommending cities as destinations. Furthermore, a lot of travel recommender systems typically rely on expert knowledge to define a representation and characterize the travel destinations in order to come up with recommendations. While this enables them to achieve a high quality of recommendations, it also drastically hinders the scalability of the systems [30, 61, 47].

Another important aspect of recommender systems revolves around the elicitation of user preferences. This is especially challenging in the tourism domain, as the tourism items to recommended are usually complex to characterize and the users may not always be able to tell their preferences to the system in a clear way. In fact, they might not explicitly know these preferences themselves [47]. Given that the accuracy of recommendations also depends on how well the system understands the user preferences, it is particularly worthwhile to investigate what viable preference elicitation methods are appropriate when building a travel recommender system. Conversational style interaction and critiquing have been suggested in previous studies as a way to address preference elicitation and enrich the user's experience with the recommender system [51, 3]. These methods work through gradually refining the user profile and eliciting user preferences through multiple interaction steps [20]. However, this is not a one-size-fits-all solution for preference elicitation and it is important to evaluate how it affects the perceived quality of recommendations before employing it in a travel recommender system.

Research Questions

Based on our motivation, we derive the following research questions in this thesis:

RQ1 How can we characterize travel destinations in a scalable way, without having to rely on expert knowledge? What features need to be considered and what representation model is suitable to use?

RQ2 How does conversational style of interaction affect the perceived quality of recommendations? Can recommender systems which employ critiquing in their interaction style achieve higher perceived quality of recommendations compared to systems that use traditional methods of interaction?

To answer our first research question, we study approaches that could be used to reduce the reliance of travel recommender systems on expert knowledge. We propose a scalable approach to characterize travel destination based on inferring information from single POIs. We acquire data about POIs for approximately 200 cities and integrate data sources for cost and climate information, to come up with a representation for the cities. Cluster analysis is applied as a way to validate the results of the characterization approach. Furthermore, we build a prototype recommender system for cities based on the data generated by the characterization approach. We conduct a user study to evaluate the performance of the system as a whole and help us further examine if our proposed characterization approach is a viable method to be used for a travel recommender system.

A further contribution of this thesis is our investigation on whether critiquing can help travel recommender systems to achieve a higher quality of recommendations. We used our city recommender prototype to test this through employing A/B testing in our conducted user study. Subsequently, we did a statistical analysis of the user study results to infer whether critiquing helps in achieving a higher quality of recommendations.

The remainder of this thesis is structured as follows: Chapter 2 contains an overview of fundamental concepts of recommender systems and an analysis of related work. In Chapter 3 we present our proposed approach for travel destination characterization. We describe CityRec — our prototype recommender system for cities — in Chapter 4, and evaluate it in Chapter 5 using a statistical analysis of the user study results. Lastly, we summarize our conclusions and point out potential directions of future work in Chapter 6.

2. Related Work

In this chapter we have a look into recommender systems in the area of tourism in order to create an overview of existing solutions, their approaches, as well as their limitations. We then turn our attention to destination characterization, a process that aims to derive information about the cities we want to recommend. The method proposed in this thesis tries to accomplish this task without relying on expert knowledge, and involves inferring information about cities from single POIs; therefore, we analyze popular venue information APIs that we could use in this process. The intent of this chapter is to provide a brief summary of existing research in the domain and help further motivating the research questions of this thesis.

2.1. Tourist Recommender Systems

Borrás et al. [11] introduce a categorization of tourism recommender systems based on their functionalities. In their analysis, four broad groups of approaches emerge: *i)* suggestion of a destination and construction of a whole tourist pack, *ii)* recommendation of suitable attractions in one specific destination, *iii)* design of a detailed multi-day trip schedule and *iv)* social aspects. For the purposes of this section we are not interested in the social aspects category, so we consider a slightly simplified version of that categorization: *i)* systems that recommend a ranked list of POIs, *ii)* systems that recommend a route plan (i.e., suggest a sequence of attractions to visit) and *iii)* systems that recommend destinations and tourist packs (a coherent bundling of products). The research discussed below is not meant to be a comprehensive survey of existing tourism recommender systems, but rather a brief overview of approaches that are typical examples of the respective categories.

Recommending Ranked Lists of POIs

SigTur is a recommender system which helps managing tourism activity in complex tourist regions consisting on a wide range of tourist products and territorial features dispersed in space. Initially, user preferences are learnt through brief questionnaires, which help the system to determine an ordered list of activities in the city to recommend to the user. The system employs a diversification mechanism that aims to widen the range of suggested activities. Furthermore, SigTur also makes use of tourism domain ontologies to help the system better match the users' preferences [10].

SMARTMUSEUM, on the other hand, is an example of a more complex mobile recommender system, which focuses on context-awareness. It can detect whether the user is indoors or outdoors — based on location — and adjust recommendations accordingly. For example, when outdoors, it can display recommendations on the map, while indoors it can show relevant museum artifacts nearby. The authors make use of semantic data representation, search result clustering, and context data to improve the recommendation performance [54].

Recommending Route Plans for POIs

When visiting a city or region, tourists need to know not just a list of POIs that fit their preferences, but also a route to follow through several attractions. The reason is that they might not be able visit all available POIs given their time and budget constraints.

INTRIGUE (INteractive TouRist Information GUidE) is a personalized recommender system tailored for the city of Torino, Italy. It offers scheduling functionality which allows composing tours while taking into account preferences of heterogenous tourist groups and other constraints. The system is also capable of providing explanations for its recommendations [2, 1].

Laß, Wörndl & Herzog introduced TourRec [37], a mobile recommender system, which uses a multi-tier web service to recommend tourist trips composed of multiple POIs. The systems makes recommendations based on specified constraints and on user's rating over different categories like *Food* and *Outdoors & Recreation*. The next iteration of TourRec [36] was augmented to use context data like *time of the day* and *previously visited POIs*, and is shown to improve performance over the previous version.

Recommending Destinations and Tourist Packs

ItchyFeet is an example of such a system. It bundles destinations and purchasing services related to the trip and then presents the recommendations to the user. In ItchyFeet, the software agents reside in a multi-agent system and user requests are handled by querying internal and external sources as well [57].

Herzog and Wörndl [30, 61] developed a recommender system for composing personalized continental travels. After the user submits his/her interests e.g., *nature & wildlife*, *beaches* or *winter sports*, the systems generates a recommendation consisting on a set of regions while respecting specified time or monetary constraints. The underlying problem for picking the regions is a variant of the orienteering problem [59] using the Oregon Trail Knapsack Problem [13] as a scoring function. The region characterization is based on manually generated models by expert knowledge.

A discernable pattern in existing tourism recommender systems is that most of these systems focus on suggesting POIs — assuming that tourists already chose the destination. The approach presented in this thesis instead focuses on recommending destinations — namely cities — therefore complementing prior work.

2.1.1. User Interface Aspects

Besides the delivery of accurate recommendations, interfaces play an important role in designing recommender systems. Therefore, it is often of interest to pay special attention to interface decisions, especially considering that a lot of research in this aspect is yet to be done [21].

Most of tourism recommender systems tend to go with a web-based interface and/or a mobile specific interface. Figure 2.1 shows the distribution of interface types among existing systems surveyed by Borrás et al. [11].

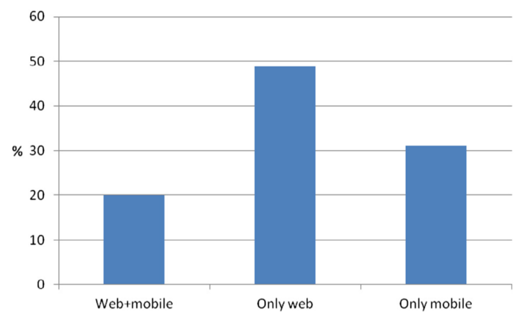


Figure 2.1.: Distribution of interface types (from Borrás et al. [11])

Although web-based approaches are still dominant, there has been a significant shift of recommender system applications towards mobile platforms, due to mobile computing devices becoming more and more ubiquitous [11, 24].

From a compatibility perspective, web-based approaches tend to be easier to develop and maintain, as they work on multiple platforms and screen sizes with minimal adaptation, while mobile-based approaches typically require separate versions for each platform. In terms of reach, web-based approaches attain wider audience by targeting desktops, laptops, mobiles, and other devices while mobile-based approaches are typically accessible in fewer environments. On the other hand, mobile-based approaches offer much better functionality in offline scenarios. Another important advantage of mobile-based approaches is that they allow for much broader personalization possibilities.

In the context of tourism systems, web-based interfaces allows users to have rich interaction with the system and typically more useful to tourist during the trip planning stage. Mobile interfaces are typically more useful to tourists during the stay period. They usually exhibit a limited interaction experience and serve less information [11]; however mobile platforms leverage opportunities for improved recommendations by allowing to personalize and contextualize the gathered information, e.g., by taking user location into account [11, 24].

Since the approach proposed in this thesis targets assisting users to find a destination, this process typically takes place during the trip planning stage. Hence, we favor the usage of a web interface in our recommender system.

2.2. Recommender System Paradigms

The recommender systems mentioned in the previous chapter utilize several recommendation techniques and paradigms to generate suggestions, which are useful to their users. In essence, recommender systems have similarities to information retrieval systems, but are differentiated in important ways: in the latter a result is no more that a match of the user's query, while in the former results are tailored according to user's preferences and viewed as options worthy of consideration [15, 52].

Burke introduced a taxonomy for recommendation techniques which has now become a widely accepted way of distinguishing between recommender systems and referring to them [14, 15]. Similar ways of categorization are also introduced in other works, too [45, 40]. In Burke's publications [14, 15], these classes are distinguished:

Content-Based Filtering The system evaluates the relationship between a single user and the descriptions of the available items. Various similarity metrics are used in this process, which typically produce scores that quantify the similarity between the user and the items [38].

Collaborative Filtering In this technique target users are recommended items similar to those liked by other users with similar preferences in the past. The preference similarity of two users is calculated based on their rating history. One of the earliest implementations of this technique was introduced by Goldberg et al. in 1992 [26]. Collaborative filtering has since remained a very popular technique used widely in e-commerce and social media, among others [12].

Knowledge-Based Filtering These systems try to reason about what items fulfill the user's requirements through using a knowledge-based approach. A knowledge base is typically build through surveying user actions or by directly asking questions about preferences. A similarity function is used to estimate how well the user's needs match the candidate recommendation items [17].

Demographic Filtering These systems rely on the demographic data of the user, e.g., age, gender, country of origin or level of studies. The recommendation is not based on the users' interests and preferences but on their personal characteristics instead. This technique is primarily used in marketing [11].

Hybrid Recommender Systems Often single techniques have inherent shortcomings, such as the cold start issue in collaborative filtering approaches [28]. Therefore many systems use a combination of the aforementioned techniques rather than applying a single one. This allows to leverage the features of one technique while compensating for the shortcomings of another [15].

The survey by Borrás et al. shows that content-based techniques are very popular among tourism recommender applications and are often used in combination with other techniques [11]. We cover more details about these paradigms in the next subsection.

2.2.1. Content-Based Recommendation

Content-based systems are characterized by their focus on exploiting the information in the item descriptions. These systems build their recommendations by calculating a degree of similarity between the user profile and the items to be recommended [31]. This approach requires some matching technique between the features of the items and the user preferences, and typically assumes that both users and items share a common representation [11].

There are several ways to represent content about an item. In content-based systems — which have historically been developed to filter and recommend text-based items, such as e-mail messages or news — a standard way to represent items is based on maintaining a list of relevant keywords about the item [31]. A simple approach would be to use a Boolean vector, where a 1 indicates that the keyword describes the item, while a 0 indicates the opposite. An obvious drawback of this method is that it assumes every keyword to be of the same importance. To address the shortcomings of this method, items can be also represented as vectors in a multidimensional Euclidean space — for example, by using the TF-IDF encoding format [55] — where the dimensions correspond to keywords or attributes of the items and the values correspond to the relevance of that attribute.

Content-based systems rely on having accurate knowledge of the user's preferences in order to select appropriate items, therefore, it is important to define an appropriate measure to compare a user and an item. In systems where the user profile and items are described by a list of keywords, various similarity measures can be applied, like for example the *cosine similarity* metric. This is the case in the approach by Lamsfus et al. [35] where user profiles and items are described by ontology concepts and the similarities are calculated by using the cosine similarity. On the other hand, in systems where user profile is represented as a vector of scores, the similarity is often computed through employing distance measures such as the Euclidean metric. In Turist@ [6] further score aggregation techniques are introduced to come up with more accurate estimations for the similarity. We discuss user profiles in more depth throughout the next section.

Although content-based systems often achieve high quality levels of recommendation, they do have a few drawbacks. A prominent drawback — especially in the tourism domain — is *overspecialization*: recommending only items similar to the user's profile, while leaving aside other items that might be interesting for the user [31]. To address this, often a diversification mechanism is employed [10, 54, 56], and the systems might skip some high-ranked recommendations in favour of diversity.

Content-based techniques are often the preferred choice among recommender systems in the tourism domain [11, 18]. A possible reason of this inclination towards content-based techniques is the characterization of tourism as a domain with *high-risk* nature (costs are relatively high), *low churn* (relevance and value of the items is not too volatile), and *low heterogeneity* (the needs satisfied from the items are not very diverse) [47]. Furthermore, the number of user ratings in tourism is typically lower compared to domains like movies or books, thus making collaborative filtering techniques, which are known to suffer from sparsity in ratings, not particularly suitable for the tourism domain. Considering this and the nature of the data we plan to use for recommending cities, content-based techniques fit well with the proposed approach in this thesis. As previously mentioned, collaborative filtering techniques would not be suitable in this context, as it would be problematic to find reliable user ratings for cities; while demographic filtering techniques would generate very broad, non-personalized recommendations. Therefore, the proposed approach in this thesis primarily makes use of content-based techniques to generate recommendations.

2.3. User Modeling and Preference Elicitation

User modeling is a cross-disciplinary research topic, which mainly focuses on the process of building and maintaining a conceptual understanding of the user [23]. In combination with preference elicitation, they make for a number of open issues in research, which are subject of on-going studies [52].

In the context of recommender systems, the goal of user modeling is to generate user profiles: mechanisms that store information about user preferences and make it possible to generate personalized recommendations [11]. Recommender systems typically focus on generating user profiles that closely reflect the concrete user needs. There are different approaches used for user profile models, the simplest of which associate a list of keywords or categories to each user, and classify the items to recommend accordingly. Although this method introduces a degree of personalization in the recommendation, it is usually too broad to produce achieve the desired accuracy [11]. A more popular method, which is also used in CRUZAR by Mínguez et al. [44], consists in storing a vector with ratings corresponding to the attributes of the items. The ratings refer to the degree of interest of the user for each attribute [11].

It is common to see augmentations of these models through techniques of knowledge representation in artificial intelligence; for example, through the use of ontologies like in SigTur by Moreno et al. [46] or in the more recent publication by Grün et al. [29].

In the latter, ontologies are used as means of refining the user profile and enriching the generic preferences of a tourist through more specific interests [29]. The system builds a user profile by asking the user to rate a set of predefined *travel factors*, that have previously been inferred from the concept of tourist types [47]. This can be naturally mapped to a vector-space model, where each dimension corresponds to a tourist factor (e.g., *cultural visitor*, *nature lover*, etc...), and the score indicates how much the user identifies him/herself with that certain factor, as depicted in Figure 2.2

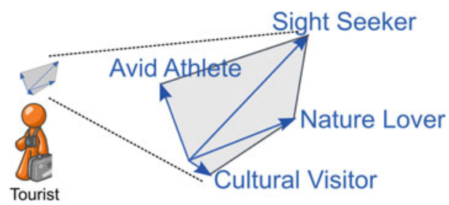


Figure 2.2.: Vector model of the user profile (from Grün et al. [29])

Exploiting the fact that the items to be recommended are similarly represented in the same space, the system uses the Euclidean metric to calculate the distance between the user and the tourist objects to generate the list of POIs to recommend. After receiving user feedback on the proposed items, the system makes use of an ontological model to generate refined profiles. To achieve this, the spreading activation algorithm [58] is applied and the tourist objects are represented based on the modularized ontology introduced by Barta et al. [5]. As soon as an object receives a rating, the set of concepts within the ontology is identified that semantically describe the corresponding object. Based on this, interest scores are assigned to the directly related concepts in the ontology and furthermore scores are inferred for the other concepts by exploiting the relationships between them in the ontological hierarchy. The user profile is thus formed by the interest scores assigned to the concepts of the ontology. The results of this process are finally combined with the initial user profile in order to achieve a more accurate representation of the user preferences.

In the recommender system described above, user feedback is collected *explicitly*, i.e., by means of direct interaction with the user. While this approach provides precise knowledge because the data is given directly by the user, it can be considered quite an intrusive way of elicitation. Users might not be keen on spending time in answering questions or filling in forms [11]. In the contrary, techniques based on *implicit* feedback aim to collect the user information by analyzing his or her behavior in the system, such as the alternatives that are selected, purchased or viewed. For example, Savir et al. [56] survey the user's modifications to adjust the budget, daily time, and maximum travel distance to collect more information that comes with no cost of additional effort from the user. However, implicit feedback does not completely replace explicit feedback, as data obtained through implicit techniques is often associated with more uncertainty than explicit information. This is reflected also in the survey by Borrás et al. [11], where 60% of the works were observed to use only explicit feedback, whereas the rest combine explicit and implicit feedback.

During the preference elicitation process, it is of importance to acquire personality information in a non-intrusive fashion [52]. In contrast to the approach by Grün et al. [29] — where users are asked to go through a self-evaluating process and need to create an understanding of the tourist types — Neidhardt et al. introduced a novel approach to acquire user preference information and build accurate user profiles [47]. In this publication the authors used the 17 tourist roles identified by Gibson & Yiannakis [63, 25] in combination with the renowned *Big Five* personality trait taxonomy [27] to come up with a reduced set of seven tourist factors that are sufficient to capture distinct tourist behavioral patterns. By asking the tourists to select a number of pictures that appeal them, the recommender system is able to quantify how much a user is represented by each factor. Similarly, as in the approach by Grün et al. [29], the user profile and the POIs are represented in the same vectors space, where the dimensions correspond to the seven factors. This enables the system to use the Euclidean metric to determine the distance between the user profile and the POIs to be recommended. The user study conducted for evaluation showed that the users were very satisfied with the system and indicated that the system is able to capture the user needs and preferences in a nonverbal way [47].

A drawback of the approaches introduced by Grün et al. and Neidhardt et al. is that the mapping of POIs onto the seven tourist factors strongly depends on experts' judgments and is rather time-consuming [47], which reduces the scalability of these systems. Addressing this issue, the authors point out the need for automated and scalable solutions.

In the approach proposed in this thesis, we favor explicit feedback techniques — which, as discussed, produce more reliable information — while aiming to find a good balance so that the preference elicitation process is not too intrusive. We also use the same vector space to represent both user profiles as well as destination cities to be recommended, thus allowing us to readily compare them and generate content-based recommendations.

2.4. Conversational Recommender Systems

Aside from the algorithms used to compute the recommendations, the mechanisms through which users provide their input and the means by which they receive the systems output, play an important role in the perceived quality of interaction with the system [52]. To better understand the implications, we now turn our attention to a popular approach for preference elicitation: conversational recommendation techniques.

In traditional methods for preference elicitation, a lot of recommender systems typically rely on explicit information collected by asking the users to specify their preferences in the beginning of their interaction with the system. Considering that the information is gathered in an explicit fashion, the inferred preferences are usually reliable, but there are a few drawbacks that can be identified in this approach: *i)* users are expected to have an understanding of the domain model used by the recommender system to identify the preferences, *ii)* users find it difficult to assess their exact preferences until dealing with the actual set of offered options, as uncertain preferences might become certain only after a significant amount of interaction with the system [49], and *iii)* users might be skeptical of the system and hesitate to reveal their complete preferences, until they see some value in interacting with the system [51].

Although implicit feedback techniques address some of these issues, the preference information gathered in the process tends to be less precise and more uncertain. Much research has been done to address these issues and one of the most prominent methods that have emerged revolves around *conversational recommender systems* [16, 41]. These systems — often also referred to as *critiquing-based recommender systems* — have broadly been recognized as an effective preference based search and recommender technology [20].

Conversational recommender systems employ *critiquing*, which helps the system narrow the search space and help the user find the item they are looking for more efficiently [42]. The user profile is built gradually, through multiple interaction steps. In each step, the system asks a question or suggests an item and the user answers the question or *criticizes* the suggestion (e.g., "I would like something cheaper"). As a result, the system incrementally refines the preference model through each step [20]. The typical interaction process is depicted in Figure 2.3.

An early adaption of this method can be seen in MobyRek by Ricci and Nguyen [51], a mobile recommender system which helps users find a restaurant of their liking. The system integrates long-term and session-specific preferences based on past user

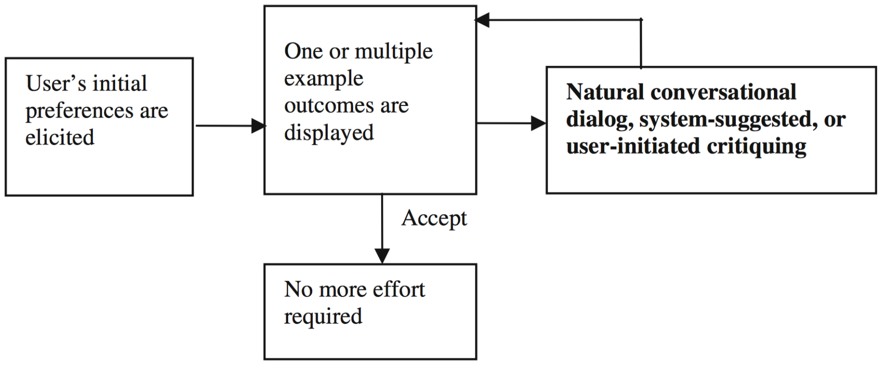


Figure 2.3.: Typical interaction process in a conversational system (from Chen & Pu [20])

interactions, and by introducing a type of critique that lets users express additional session-specific preferences alongside their strengths (must-have or nice-to-have). Results of their conducted user studies showed that the conversational style they used to elicit user preferences was very effective. The authors, however, point out that users should also be given the option to provide their preferences explicitly and not always only through critiques [51]. MobyRek's functionality was further extended by Averjanova et al. [3], with the integration of maps as the main means of interaction with the system.

A more recent example of a recommender system employing conversational interaction style can be found in CT-Planner [33] and further refined in CT-Planner4 [34]. The system recommends tour plans that are refined gradually as the users reveal their preferences such as duration, day of the week, walking speed and reluctance to walk.

Pu & Chen [20] establish a categorization of conversational recommender systems as follows:

Natural language dialog-based systems In these systems users are engaged in a conversational dialog and prompted to provide preference feedback to the current recommendation. Especially suitable for generating recommendations that need to be delivered by speech interfaces, rather than visual platforms.

System-suggested critiquing systems These systems typically pro-actively generate a set of knowledge-based critiques that users might accept as ways to improve the current recommendation.

User-initiated critiquing systems On the other hand, in these type of systems the focus is on showing examples and stimulating users to make self-motivated critiques. The main advantage of this approach is allowing a higher level of user control, but with the cost of additional user effort.

Furthermore, Pu & Chen highlight the pros and cons of these approaches and suggest potential hybrid approaches that unify the advantages and try to overcome limitations of single approaches [20]. McGinty and Reilly point out that although there has been a considerable level of research activity in the area of critiquing, there are still many open challenges and opportunities [43].

In conclusion, the conversational approach in preference elicitation is an effective and helpful method to acquire user preferences. We make use of this style of interaction in the system proposed by this thesis, with the goal of enriching the user experience and improving the system's understanding of user preferences. Furthermore, we investigate how it affects the perceived quality of recommendations by conducting a user study.

2.5. Destination Characterization

In the context of this thesis, a destination refers to a distinct geographical area to be recommended. To distinguish between the granularity levels of destinations, it's common to refer to hierarchical trees for regions; for example the hierarchy proposed by Wikitravel[1], a popular free and crowdsourced travel guide: Continents ⟶ Continental sections ⟶ Countries ⟶ Regions ⟶ More regions ⟶ Cities ⟶ Districts. In this thesis we are interested in building a recommender system for cities, which lean towards the fine grained end of this hierarchy.

Recommender systems are in essence information processing systems, therefore they heavily rely on continuously gathering various kinds of data to generate the recommendations. Depending on the used recommendation techniques, some systems base their suggestions on simple data representations, like basic user ratings, while other systems rely on more involved representations like ontological descriptions, for example. We already discussed several methods, which serve the purpose of inferring and maintaining data about users in the previous sections. In this section, we instead focus on information about the items to recommend, which in our case are cities. More specifically, we are concerned with city characterization: deriving

[1] `https://wikitravel.org/en/Wikitravel:Geographical_hierarchy`

information about the destination city, i.e., gathering and representing relevant data to enable recommendation. It is common to address tasks of this nature through expert knowledge. In the next subsection we discuss some expert knowledge approaches and their limitations, while pointing out how we plan to address this task differently in our approach.

2.5.1. Expert Knowledge in Recommender Systems

Existing approaches typically rely on expert knowledge to define a representation and to evaluate the destination items for making recommendations. A common drawback with such approaches is that they do not scale well as the number of items considered increases, and they depend on the subjectiveness of human judgement.

For example, in the recommender system for composing personalized continental travels that we discussed about previously [30, 61], a content-based approach is used for recommendation. Therefore a characterization of the items to recommend — in this case regions — is essential to the recommender system. Currently, this relies on expert knowledge, which makes it challenging to keep it up to date and hinders scalability. Messaoud et al. extend this recommender system by focusing on the diversity of activities [19] within a composite trip, through using a hierarchical technique [8]. However the approach still relies on expert knowledge. Dietz et al. propose using a data-driven technique to scale these approaches up, specifically the parts related to calculation of stay durations. The technique is based on mining spatial-temporal data from location-based social networks (LBSNs) to derive tourist mobility patterns [22]. This approach could make it possible for [30, 61] to rely less on expert knowledge.

In recommender systems proposed by Grün et al. [29] and by Neidhardt et al. [47], the recommended items are single POIs instead of regions, but the nature of the problem about item characterization remains the same. As the authors point out, the evaluation of POIs' attributes is strongly based on experts' judgments and doesn't scale well. The authors suggest the usage of text mining methods in future works to alleviate these issues in their approaches.

Bidart et al. introduced ReCWEE and ReCWEE+, which are collaborative filtering recommendation methods for suggesting a set of cities to the user [9]. The proposed method exploits information from two conceptual layers — information about the cities and information about the attractions in each city — and tries to aggregate them to improve the recommendations. The used dataset is collected from TripAdvisor, a popular travel website[2]. While this approach is potentially scalable, a drawback is that it is based on collaborative filtering, known to suffer when user ratings are sparse — as is usually the case in the tourism domain. Burke and Ramezani argue that content-based and/or knowledge-based approaches are typically fitting in the context of tourism [18].

Nomad List[3], a popular website in the travel community, maintains a large database of cities in the world. It addresses the task of city characterization in a novel way: through crowdsourcing and mechanisms to aggregate the users votes. The service provides ratings for various attributes of a city, as depicted in Figure 2.4.

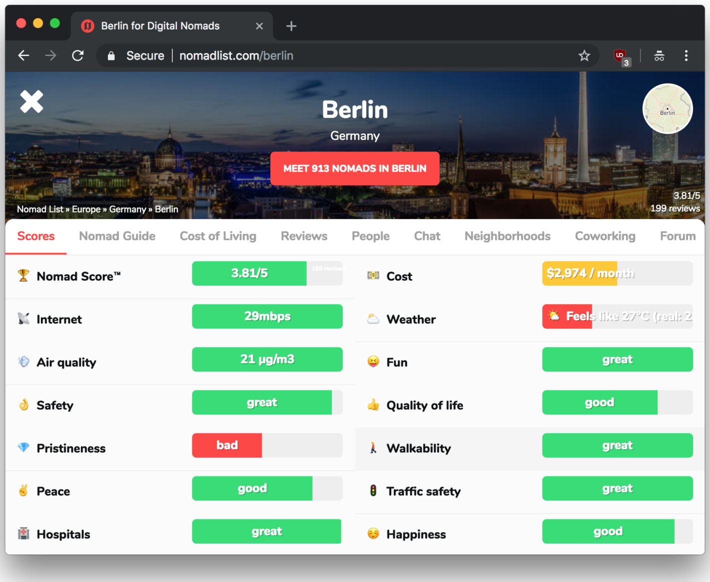


Figure 2.4.: Nomad List

In contrast with previously mentioned approaches, the recommender system proposed in this thesis aims to drastically reduce reliance on expert knowledge. We propose a scalable approach for characterizing cities based on inferring information about them from single POIs. This approach could be applied to other levels in the region hierarchy too, with minimal adjustments.

[2] `https://tripadvisor.mediaroom.com/us-about-us`
[3] `https://nomadlist.com`

2.5.2. Comparison of Venue Data APIs

To address the city characterization task, we acquire data about POIs in the cities. Thus, we analyze popular venue information APIs in the following. This comparison is not meant to be exhaustive and is predominantly based on the claims of the services themselves, as well as publicly available data. It servers as a brief survey of available venue information services at the time of writing, with the goal of finding a suitable choice for the purposes of the approach proposed in this thesis.

During the recent years, a lot of location based applications have surfaced. Alongside them, also venue information providers have emerged, primarily designed to support the business use cases of each respective provider. A prominent service is the Google Places API[4] — hosted by a major player in the industry. It returns information about places using HTTP requests. Places are defined within the API as establishments, geographic locations, or prominent points of interest. It claims to have 1 billion monthly active users and 25 million daily updates to maintain information accuracy.

OpenStreetMap[5], on the other hand, is a volunteer-driven collaborative effort to maintain an up-to-date map of the world. It hosts a variety of APIs in its ecosystem and one of the most noteworthy functions is the capability to download the entire map data collection[6]. The venues are partitioned in very granular categories, but there doesn't exist a clear taxonomy for the categories. Furthermore, it is also common to find duplicate venue entries.

Foursquare[7] is primarily a location-based social network. It also provides an API for venue information for which the data comes from over 13 billion first-party check-ins, combining machine learning models and first-party data from their consumer apps. The Foursquare database includes over 105 million points of interest and contains valuable venue information, from addresses and opening hours to ratings and need-to-know tips.

Other services like Yelp are more orientated towards event and business POIs, via its Fusion API[8]. Factual's Global Places API[9] is another venue information service. A particularly interesting attribute in their venue representations is *existence*: a machine-

[4]`https://cloud.google.com/maps-platform/places/`
[5]`https://www.openstreetmap.org`
[6]`https://planet.openstreetmap.org/`
[7]`https://foursquare.com`
[8]`https://www.yelp.com/developers/documentation/v3`
[9]`https://www.factual.com/products/global-places/`

learned numerical value between 0.0 and 1.0 indicating the confidence that the POI is real, open, and not a duplicate. This allows the service users to set a threshold and find their preferred trade-off between comprehensiveness and accuracy.

In Table 2.1, we summarize a few relevant aspects of these services. For some of these aspects we use plus symbols (+) as a method to visually compare the services relative to each other. A larger amount of plus symbols in the *Volume & coverage* aspect for example, illustrates that the service performs relatively better in this aspect. Although this is a simplification of the details, it serves the purpose of creating a general overview of the services.

Service provider	Request limit	Pricing	Volume coverage	Categorization granularity
Google Places API	No limits	Free for basic queries, pay-as-you-go pricing model for more involved queries [10]	++++	+++
OpenStreetMap	N/A [11]	Free	++++	+++
Foursquare API	99,500 regular & 500 premium API calls per day [12]	Free within the API call quota, commercial license starting at $599 per month	++++	++++
Yelp Fusion API	5,000 API calls per day	Free within the API call quota [13]	+++	++
Factual	N/A [14]	Not made public	++	+++

Table 2.1.: Comparison of venue information APIs

In the approach proposed in this thesis, we are primarily interested in the categories of the venues of a city, to infer knowledge about the city. The Foursquare API provides a rich and well-structured taxonomy of the venue categories. This service also allows dispatching a substantial amount of requests without imposing many limits. Considering this, and the fact that Foursquare does have good coverage of venues in a myriad of cities, we opt for using the Foursquare API to acquire POI data. In the later chapters, we explain how the system makes use of this data to recommend destination cities to the users.

[10] `https://developers.google.com/maps/billing/understanding-cost-of-use#data-skus`
[11] Does not apply, as the data is first downloaded locally
[12] `https://developer.foursquare.com/docs/api/troubleshooting/rate-limits`
[13] `https://www.yelp.com/developers/faq`
[14] Does not apply, as the data is first downloaded locally after the payment is completed

2.6. Conclusions

In this chapter, we analyzed some aspects of recommender systems in the area of tourism and discussed about existing solutions. We first considered the main functionalities of popular tourism recommender systems. Most of the used approaches focus on suggesting POIs, thus assume that the destination has already been selected. There seems to be a discernable absence of systems which recommend cities as destinations. Thus, the system proposed in this thesis complements prior work by focusing in destination city recommendation.

We then briefly analyzed the interfaces used by these systems and pointed out the advantages and disadvantages of web- and mobile-based approaches. Web-based interfaces remain dominant, although there is an increasing trend of using mobile interfaces. Considering that tourists typically look for destination cities during their trip planning stage, we favor the usage of web-based interfaces over mobile in our system. Exploiting recently developed web technologies, it would be possible to also adapt the interface for usage in mobile devices with minimal adjustments.

The recommendation techniques are essential to the performance of recommender systems, hence we also discussed the main paradigms behind these techniques. Due to the nature of the recommendation problem in tourism and the type of data available, content-based techniques are typically preferred in this domain. These are often combined with other techniques to address certain limitations. Since the approach proposed in this thesis aims exploiting venue information to build knowledge about the cities to recommend, we primarily utilize content-based techniques to generate recommendations.

We subsequently discussed user modeling and preference elicitation approaches. We compared the approaches used by existing systems and also treated the usage of knowledge representation techniques to augment these approaches. The conversational approach in preference elicitation was proposed in literature as an effective and helpful method to acquire user preferences. We employ this approach in our system, and investigate how it affects the perceived quality of recommendations by conducting a user study.

Finally, we treated the topic of destination characterization. A lot of existing approaches rely heavily on expert knowledge to address this task, thus reducing the scalability of the systems and introducing a dependency on the subjectiveness of human judgement.

3. Travel Destination Characterization

In this chapter we describe our proposed approach to characterize travel destinations by aggregating information from single POIs. Initially, we discuss the process of venue data acquisition and its challenges. Using the newly acquired information about the POIs, we then analyze possible methods to determine the features and scores for each travel destination in the dataset. Finally — as a way to validate our findings — cluster analysis is used to determine whether there is any discernible group structure among the characterized travel destinations.

3.1. Venue Data Acquisition

In the literature review, we briefly compared various API providers for venue data, and concluded that Foursquare's API was the most suitable for our approach. Mainly because of its rich, well-structured taxonomy of the venue categories and also the reasonably generous API rate limits. The API exposes various endpoints to its users, which serve different use-cases, e.g., search venues, get venue recommendations, get similar venues, get information about users, get information about check-ins and numerous others. We are interested in the search venues endpoint[1], which allows us to specify coordinates of a bounding box in the request parameters, and returns information about the venues within the specified area of the map. The returned venue information includes elements like name, contact details, location details, assigned categories from Foursquare's taxonomy, statistical data, as well as other less relevant details.

One important limitation of this endpoint is that it returns only up to 50 venues per query, even if the area specified by the bounding box has more venues to offer. This limitation makes the process of acquiring venue data non-trivial, i.e., we cannot simply specify a bounding box that encompasses a large area (for example a whole city) and

[1] https://developer.foursquare.com/docs/api/venues/search

expect to receive all the available venues of that area in the query response. Considering that for a given area, the total venue count and venue density are not known a priori, the necessity of a more involved data acquisition approach becomes clear. We discuss such an approach in the next section and describe the arising challenges and objectives.

3.1.1. Algorithm to Collect Foursquare Venue Data

The objectives of such an algorithm are straightforward:

- Collect all available venues of the area specified by the bounding box, and
- Minimize the amount of API requests used,

while the most relevant limitations that the algorithm has to take into consideration are:

- The API returns only up to 50 venues per request.
- The total venue count and venue density of a given area, are not known a priori.
- API request count and rate are limited by the provider.

A naive approach to this problem would be to simply split the area defined by the bounding box into a grid of rectangles and query the API for each of these rectangles separately. The problem with this solution is that it assumes the specified area to have the same venue density across its surface, thing which rarely holds true in a real scenario. For example, cities — the travel destinations we are interested in — typically have a large difference in venue density between the areas close to the city center and the areas in the outskirts. Another issue with this solution is the selection of the grid size. A lower grid size would split the specified area into fewer rectangles, hence utilize fewer requests but potentially miss out collecting many venues. While a higher grid size would result in a better chance of collecting all available venues, but a lot of the API requests potentially go in vain, as the low-density areas of the grid would needlessly utilize a large number of API requests.

The method we used is based on a straightforward recursive approach. The steps are briefly described in Algorithm 1, in the form of high-level pseudocode.

Using this recursive approach allows to skip querying the areas with no venues, and

Algorithm 1 Venue collection approach

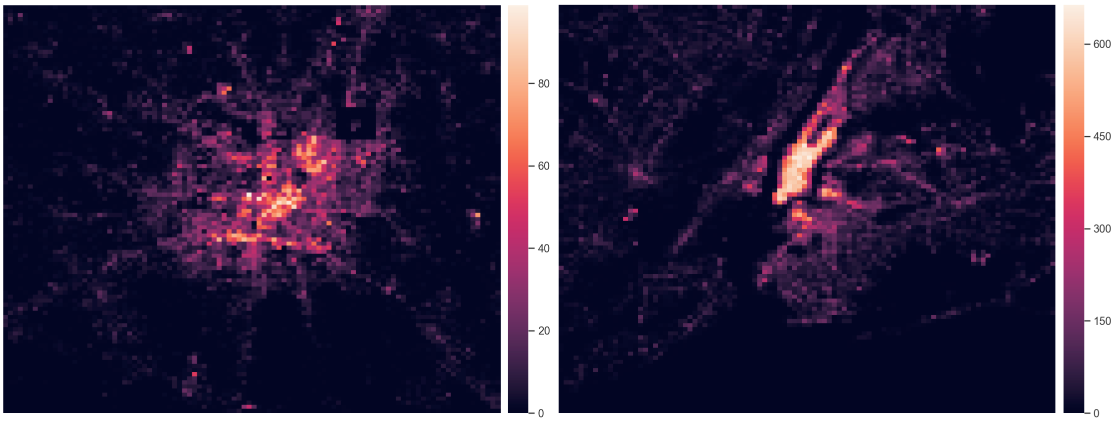
```
procedure GetCityVenues(CityBoundingBox)
    UniqueVenues ← ∅                    ▷ Create a reference to a set of unique elements.
    InitialGridSize ← 20
    GetAreaVenues(CityBoundingBox, InitialGridSize, UniqueVenues)
    return UniqueVenues
end procedure

procedure GetAreaVenues(AreaBoundingBox, GridSize, UniqueVenues)
    GridTiles ← PartitionArea(AreaBoundingBox, GridSize)        ▷ This procedure
  splits the given area to a grid with GridSize^2 tiles. Pseudocode for this procedure is
  omitted in favor of brevity.
    for Tile ∈ GridTiles do
        Venues ← getFoursquareVenues(Tile)
        UniqueVenues ← UniqueVenues ∪ getFoursquareVenues(Tile)
        if |Venues| ≥ 50 then
            GetAreaVenues(Tile, 2, UniqueVenues)              ▷ Make recursive call.
        end if
    end for
end procedure
```

therefore spare a large amount of API requests. Furthermore, we are certain that all the venues of the specified area are collected, as we recursively divide the area to smaller parts and query them separately until the returned number of venues is smaller than 50. Meeting this base case condition implies that the area is small enough to be scanned in a single query.

The efficiency of the method described in Algorithm 1 can further be improved by tuning parameters like *InitialGridSize* and *GridSize*. Additionally, a few quirks of Foursquare's API need to be handled as well. We consider these to be implementation details and therefore choose not to discuss them in this section.

Figure 3.1 shows the distribution of the collected venues across the city area for two sample cities: Milan and New York. As expected, we can observe the widely different levels of venue density across the city areas. Applying our proposed method to collect venues for Milan results in 51327 total venues and consumes approximately 3100 API calls. While applying it for New York results in 286448 total venues and consumes approximately 21000 API calls. For comparison, the naive method we previously



(a) Milan (b) New York City

Figure 3.1.: City venue distribution on a 100x100 grid. The heatmap illustrates the widely different levels of venue density across the city area.

mentioned would consume 10000 API calls for Milan (100x100 grid), and 40000 API calls for New York (200x200 grid), while still failing to collect all available venues.

3.2. Characterizing Travel Destinations Based on Venue Data

We now turn the attention to methods that allow us to characterize the travel destinations based on venue data collected from Foursquare. In this thesis, cities are the travel destinations of interest, hence we continue the discussion about the characterization approach with cities as the point of reference. However, the approach that we propose could potentially also be used for characterizing destinations of different granularity levels.

Using the methodology described in the previous section, we collected venue data of approximately 200 cities around the world. The cities were handpicked based on tourist guides and online searches for popular tourist destinations. The complete list of the cities we selected can be found in Appendix A.

In order to establish an association between the cities and the venues, we look into the *category* feature of the venues in the dataset. Conveniently, Foursquare provides a well-defined venue category hierarchy[2] which allows us to map every venue to a top

[2] https://developer.foursquare.com/docs/resources/categories

level category, e.g., Science Museum $\longrightarrow$ Museum $\longrightarrow$ Arts & Entertainment. We are especially interested in the top level categories, which are:

- Arts & Entertainment
- College & University
- Event
- Food
- Nightlife Spot
- Outdoors & Recreation
- Professional & Other Places
- Residence
- Shop & Service
- Travel & Transport

We use a subset of these categories — which are more relevant to tourism — to create a feature set that enables us to characterize the cities, namely *Arts and Entertainment*, *Food*, *Nightlife*, *Outdoors and Recreation*, *Shops and Service*. These features can be conceptualized as a multi-dimensional vector space model where each feature corresponds to a dimension and each city is represented by a point in this space. To fulfill the characterization task we now need find a method to score all the cities along each dimension of the aforementioned vector space. These scores would ideally quantify how much the city is associated with each feature. For example, based on the scores, one would be able to distinguish cities which are more oriented towards *Outdoors and Recreation*, from cities oriented mainly towards *Food* and *Nightlife*. In the next section, we discuss our proposed approach to estimate these scores based on the venue data.

3.2.1. Scoring Cities Based on Venue Category Distribution

We use a rather simple approach to estimate the city scores, which is based on its venue category distribution. By dividing the number of venues per each top level category by the total venue count of the city, we obtain the percentages of each top level category in the city's category distribution. Our approach mainly relies on the assumption that these percentages can be used as indicators for the association level of the city with the features corresponding to the top level categories. A simple example helps demonstrating this: the cities in our dataset have a certain distribution of venue categories; if the number of venues labeled with "Arts & Entertainment" in a city is on the high end of that distribution, it can be assumed that the artistic experience of that city is better compared to a city on the other end of the distribution. Figure 3.2 illustrates the category distributions of a few cities.

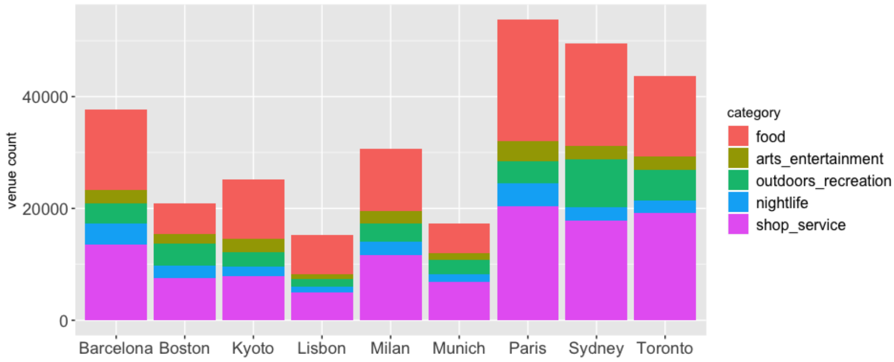


Figure 3.2.: Venue category distribution for a subset of cities

By working with percentages instead of venue counts per category, we eliminate the effect of the city size in our features. However, looking at Figure 3.2 we can see that in many cities have a somewhat similar distribution of venue categories, where "Food" and "Shops & Service" , unsurprisingly, dominate in general. We want the feature scores to represent the rank a given city relative to other cities in the dataset, therefore we apply min-max scaling to the calculated percentages. This way we obtain the final city scores for each of the features, which take values from the segment $[0, 1]$.

3.2.2. Additional Feature Engineering

Considering that the cities dataset produced by the characterization process will ultimately be used in a recommender system, we are interested in expanding the feature set used to describe the cities, possibly by looking into other tourism-related aspects such as price level and climate.

We collected climate data for our selected cities from an online source called Climate-Data[3]. Given the nature of the characterization task, we are interested in long-term aspects of the cities' climate (e.g., yearly average temperature), rather than short-term

[3]https://en.climate-data.org

aspects (e.g., current temperature). Using this data, we expand our dataset with two more features: *Average (Yearly) Temperature* and *Average (Yearly) Precipitation*.

Numbeo[4] is a very popular public database for cost of living data about cities and countries world wide. They maintain a publicly available list of more than 500 cities alongside their cost of living information. We are particularly interested in "Cost of Living Index", which is a relative cost indicator calculated by combining metrics like consumer goods prices, restaurants, transportation and others[5]. We can use this data as a proxy for price level information about the cities. This way, we add another feature to our cities dataset called *Cost of Living Index*.

Furthermore, we can derive two additional candidate features from data we already have about the cities, specifically *Venue Count* and *Area*. These features could improve our characterization model by accounting for the size of the cities, besides other aspects.

3.2.3. Feature Selection

So far, these are the features we have in our city dataset:

1. *Food* - corresponds to venues labeled with Foursquare's "Food" category
2. *Arts and Entertainment* - corresponds to venues labeled with Foursquare's "Arts & Entertainment" category
3. *Outdoors and Recreation* - corresponds to venues labeled with Foursquare's "Outdoors & Recreation" category
4. *Nightlife* - corresponds to venues labeled with Foursquare's "Nightlife Spot" category
5. *Shops and Service* - corresponds to venues labeled with Foursquare's "Shops & Service" category
6. *Venue Count* - number of total venues in the city
7. *Area* - area of the city based on official city bounds
8. *Cost of Living Index* - cost indicator based on Numbeo's statistics
9. *Average Temperature* - yearly average temperature in the city (in degrees celsius)
10. *Average Precipitation* - yearly average precipitation in the city (in millimeters)

[4]https://www.numbeo.com
[5]https://www.numbeo.com/cost-of-living/rankings.jsp

Upon further inspection, we observe that Foursquare's "Shops & Service" category contains many subcategories which do not fit well in the tourism theme. For example, "Auto Dealership", "Astrologer", "Betting Shop", "Business Service", "Car Wash", etc... Based on this insight, we choose to drop the *Shops and Service* feature.

To make sure that the features we ultimately select contribute towards determining an appropriate characterization of the cities, we apply a feature correlation analysis. Furthermore, this procedure serves as a data preparation step prior to applying cluster analysis on the dataset.

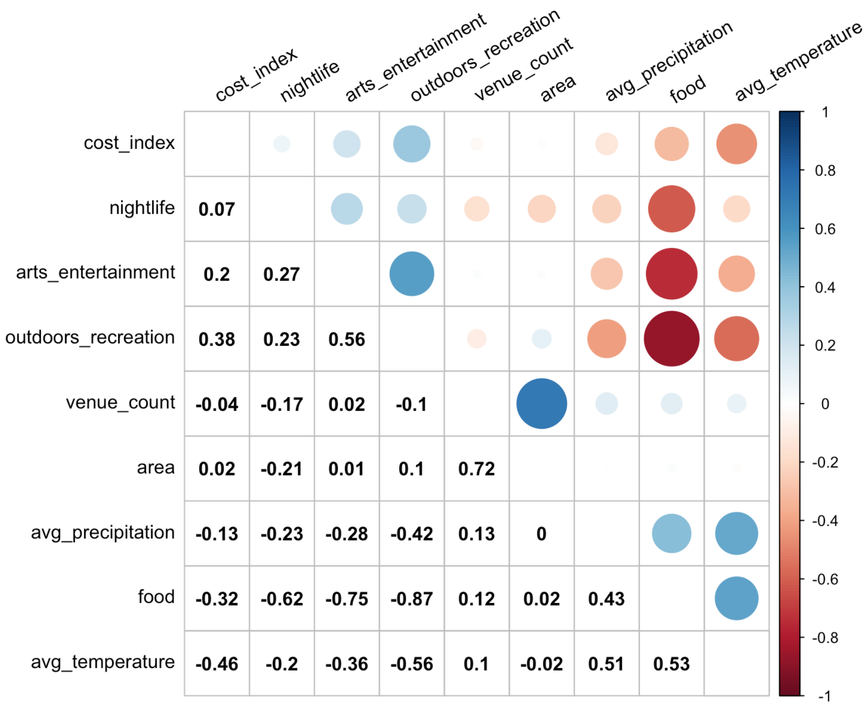


Figure 3.3.: Feature correlations

Figure 3.3 shows the correlation between the remaining features in the dataset. We notice a very strong positive correlation between *Venue Count* and *Area* with a coefficient of +0.72; this is to be expected as both capture more or less the same information about the city, namely its size. Thus, we drop the *Area* feature as it is based on official municipality/city bounds and as such, it potentially overestimates systematically the true area of the cities. Other interesting correlations can be seen between *Cost of Living Index* and *Average Temperature* (-0.46), *Food* and *Average Temperature* (+0.53), *Outdoors* *and Recreation* and *Average Precipitation* (-0.42). The features mostly exhibit medium correlation coefficients, which is a good sign, as this coefficient value range is desirable for the cluster analysis to find structure in the data. Therefore, we choose to keep the remaining features.

Eventually, we are left with eight features in our dataset: *Arts and Entertainment*, *Food*, *Nightlife*, *Outdoors and Recreation*, *Cost of Living Index*, *Average Temperature*, *Average Precipitation* and *Venue Count*.

3.3. Cluster Analysis

Considering that there are approximately 200 cities around the world in our dataset, it is natural to expect distinct groups of cities emerge from the data. Therefore, we apply cluster analysis as means of testing the validity of results produced by the destination characterization approach we used. If distinguishable and interpretable city clusters emerge in the cluster analysis, we have stronger evidence that the proposed characterization approach is appropriate for the task.

Cluster analysis is a type of unsupervised learning scheme whose goal is to group the instances of a dataset in such a way that instances within the same group are similar to each other, while the groups are dissimilar between themselves. It is often applied on unlabeled datasets to explore possible natural partitions of the instances. Popular clustering algorithms include k-means, k-medoids, and hierarchical clustering.

Considering that the features in our dataset have widely different value ranges, it is necessary to normalize the features, so that the clustering results aren't biased. Hence, we apply min-max scaling to all the features in the dataset.

A noteworthy challenge when working with clustering algorithms revolves around finding an appropriate number of clusters. We explored various cluster numbers and experimented with the clustering algorithms mentioned above. To evaluate the quality of the resulting clusters we looked into standardized metrics like within-cluster sums of squares and the average silhouette width [53]. The former is a measure of the variability of the instances within each cluster, while the latter is a measure of how well the instances fit into their assigned cluster, as opposed to all the other clusters. Based on these quality metrics, we found out that the clustering algorithms tend to perform well on our dataset when the cluster number is set to a value between two and five.

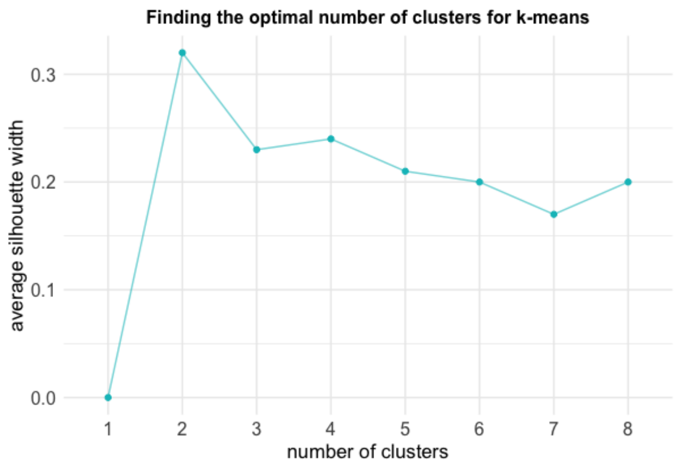


Figure 3.4.: Average silhouette widths for different number of clusters (k-means)

Using k-means as the clustering algorithm and Euclidean distance as the distance measure, we observe promising average silhouette width values. As illustrated in Figure 3.4, the average silhouette width value is about 0.3 for two clusters and remains close to 0.24 for three to four clusters; for five clusters and beyond the average silhouette width drops. Considering that the instances in our dataset are cities, a number of clusters larger that two would be desirable. Based on the reported average silhouette widths, four would be a reasonable number of clusters for this configuration.

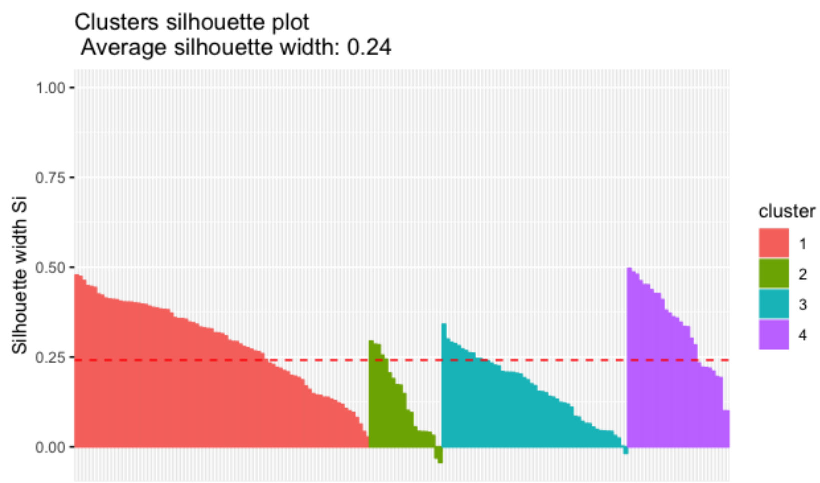


Figure 3.5.: Silhouette plot for four clusters (k-means)

Figure 3.5 shows the silhouette plot for these four clusters, which mostly lie in the positive area. This helps asserting that the resulting clusters are reasonably good. Using the within-cluster sums of squares metric to determine the number of clusters would also moderately support selecting four clusters, in alignment with the observations from the silhouette method.

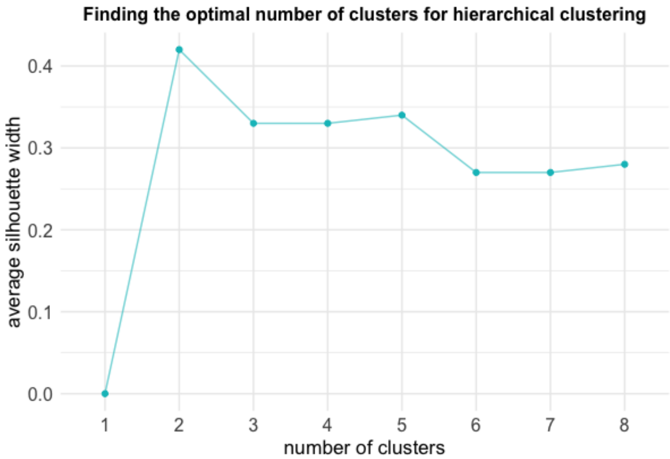


Figure 3.6.: Average silhouette widths for different number of clusters (hierarchical clustering)

After further experimentation, we found that using hierarchical clustering and Pearson distance[6] leads to higher average silhouette widths. We use the same method as described earlier to find the optimal cluster number for hierarchical clustering. As shown in Figure 3.6 the average silhouette width is approximately 0.4 for two clusters, steadily stays close to 0.34 for three to five clusters and drops to around 0.27 for six clusters and beyond. Based on these reported values, we select five as the number of clusters. Figure 3.7 shows the silhouette plot for these five clusters, where the majority of the clustered instances fall in the positive area. As in the previous case, this observation helps asserting that the resulting clusters are fairly good.

3.3.1. Interpretation of the Clustering Results

The five discovered city clusters represent the distinct groups in the dataset resulting from the city characterization process. In this section, we briefly describe the

[6]The Pearson distance is a correlation distance based on Pearson's correlation coefficient. It lies in the segment $[0, 2]$ and measures the linear relationship between two vectors.

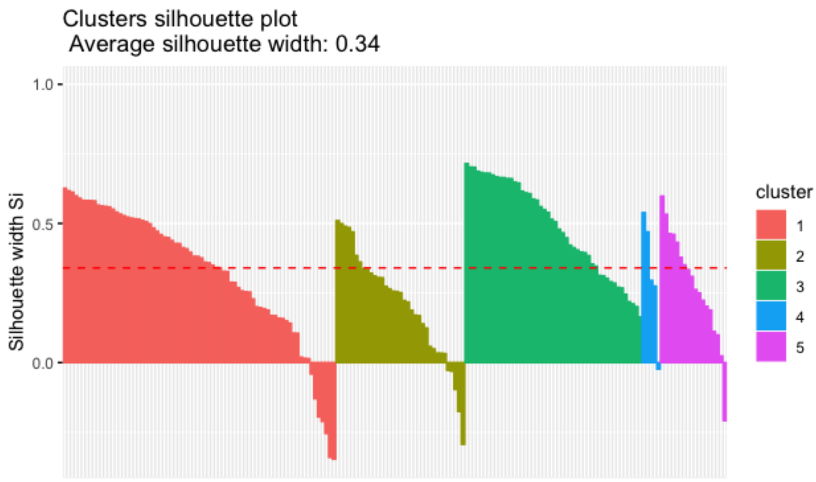


Figure 3.7.: Silhouette plot for five clusters (hierarchical clustering)

distinguishing characteristics of each cluster. Table 3.1 shows the mean values of the city features per cluster while Figure 3.8 helps visually illustrating the differences of the value distributions per cluster.

Feature	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
Instance count	74	35	48	5	18
Share	41%	19%	27%	3%	10%
Silhouette width	0.32	0.21	0.49	0.31	0.28
Food	0.35/0.12	0.58/0.11	0.73/0.14	0.62/0.19	0.41/0.08
Arts and Entertainment	0.41/0.14	0.26/0.11	0.18/0.11	0.33/0.19	0.36/0.19
Outdoors and Recreation	0.59/0.13	0.38/0.16	0.24/0.13	0.31/0.13	0.36/0.11
Nightlife	0.39/0.13	0.30/0.10	0.22/0.12	0.25/0.15	0.61/0.20
Venue Count	10907/14053	21691/31944	13081/14371	153385/42216	11446/13413
Cost of Living Index	67.67/16.62	75.77/8.70	42.90/12.69	42.28/7.38	45.36/9.26
Average Temperature	10.28/4.07	15.10/2.6	22.49/4.67	20.84/6.59	20.33/5.39
Average Precipitation	777.73/277.13	931.54/384.49	1323.35/730.67	1199.40/509.25	1062.94/448.50

Table 3.1.: Mean and standard deviation values of city features per cluster

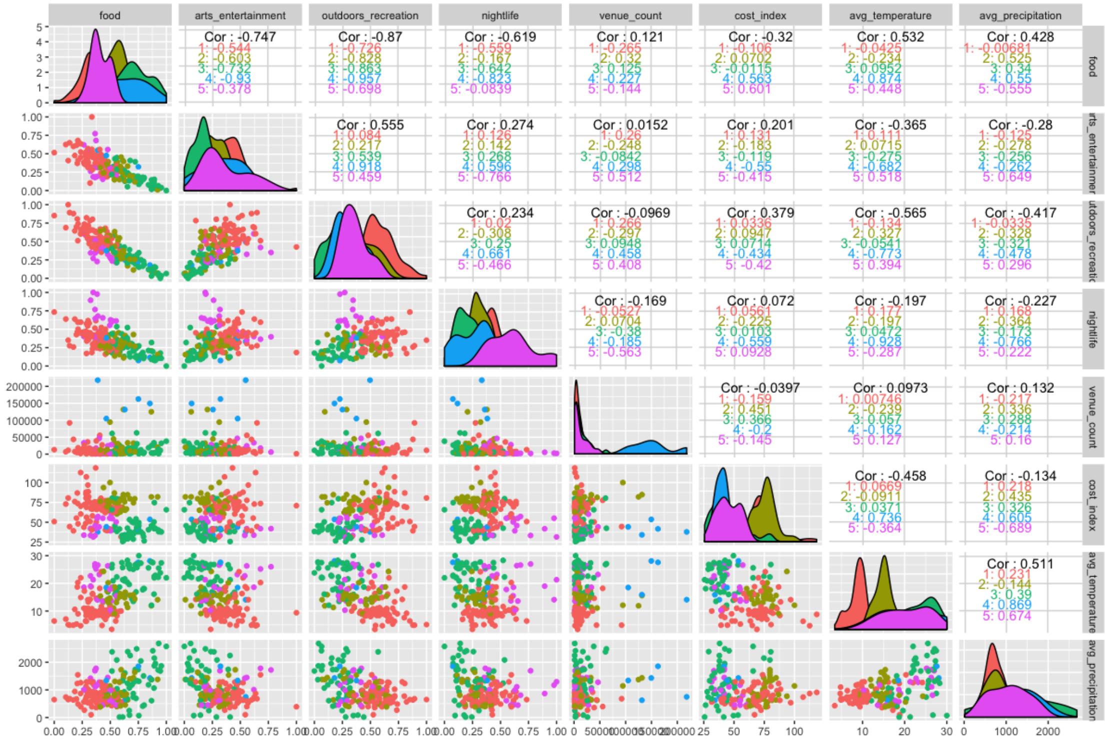


Figure 3.8.: Distribution and feature correlations of the five clusters

The majority of the cities fall into clusters 1 and 3, which have 41% and 27% of the instances respectively. Clusters 2 and 5 are of medium size, respectively containing 19% and 10% of the cities. While cluster 4 is the smallest with only 5% of the cities.

Cluster 1 and cluster 2 are quite similar to each other. Both contain cities with relatively high cost of living indices and significantly lower average temperatures than the other clusters. In cluster 1, we find a lot of cities with continental climate or similar, located in north and central Europe, like Munich, Vienna, Berlin, Amsterdam, Dublin, Copenhagen but also cities from farther away locations, like Chicago in the US, Montreal, and Toronto in Canada. *Arts and Entertainment* and *Outdoors and Recreation* are two characteristic features for this cluster, and the mean scores along these features are the highest among all clusters. In contrast, cluster 2 boasts higher scores in *Food* and has a considerably higher mean of temperatures (15.1 degrees Celsius). This is rather unsurprising, when considering that cluster 2 contains a myriad of mediterranean climate cities from southern Europe, like Rome, Nice, Florence, Verona, Marseille, and Athens. Similarly to the first cluster, this cluster has some diversity in locations too.

Some other members of this cluster include cities like Melbourne, Sydney, Kyoto, Tokyo and New York City. The venue count mean of cluster 2 is almost twice as large as that of cluster 1, hinting that cities of cluster 2 are on average larger than those of cluster 1; however, the standard deviation of this feature is very high and thus this difference of the means is not very interpretable.

Cluster 3 contains an intriguing mixture of asian and latin american cities. This cluster has a remarkably high mean value of scores in *Food* and the highest mean of average temperatures among the clusters (22.49 degrees Celsius). The cost of living index is on average 42.9 which is relatively low in comparison. Some example member city of this cluster include Penang, Siem Reap, Cancún, Chiang Mai, Medellín, Delhi, Denpasar and Quito.

Lastly, cluster 4 and 5 also exhibit similar scores along some of the features. Both are small clusters and contain cities with low cost of living indices on average (42,28 and 45,36 respectively). Both clusters also share similar relatively high values for the mean of average temperatures, hinting at cities with warm climates. The distinguishable feature for cluster 4 is its venue count. This cluster has a noticeably high mean of venue counts, which is several times larger than that of the other clusters. The cities of this cluster — Mexico City, São Paulo, Jakarta, Bangkok, and Istanbul — are also characterized by high scores in *Food*. On the other hand, the differentiating characteristic of cluster 5 is *Nightlife*. This cluster boasts the highest mean score for *Nightlife* among all clusters. Rio De Janeiro, Seville, Punta Cana, Lagos, Madrid, and Fortaleza are a few sample member cities of this cluster.

4. CityRec - A Prototype Travel Recommender System for Cities

Now that we have defined a method how to address the destination characterization task, and thus answered our research question **RQ1**, we shift our attention to creating a city recommender system based on our characterized cities dataset. In the literature review, we saw that a majority of tourism recommender systems focus on suggesting POI and assume that the travel destination is given. CityRec, the recommender system proposed in this thesis, complements prior work on tourism recommender systems by addressing an earlier step of travel planning: choosing a destination. We first discuss our proposed recommendation strategies — the core of our recommender system — and subsequently focus on more technical details about the prototype, like system design, user interaction, interface, and lastly implementation remarks. In Chapter 5 we describe how we used this prototype to test some hypotheses stemming from the research questions of this thesis, and to evaluate the perceived quality of the recommendations generated by the system.

4.1. Recommendation Strategies

In this section describe our proposed approach to recommend cities to the users. Considering that we want our recommender system to generate personalized recommendations, we first need to define a method to model user preferences and infer the user profile.

4.1.1. Inferring User Profiles

In Section 3.2, we discussed the characterization of cities and defined the features to be used in the dataset. Every city is represented as a point in a multi-dimensional vector space, where the dimensions correspond to the features. We can use the same method to represent the user profiles. This would make it possible to use distance metrics to calculate the distance between cities in the dataset and any given user profile, thus enable us to generate content-based recommendations. The feature scores in the user profile reflect the city characteristics that fit the user preferences.

To obtain the initial scores for the user profile, we ask the user to select the cities which best reflect his/her preferences from a list of 12 cities during the preference elicitation step. Using this non-intrusive way of collecting user preferences, we can build an initial user profile by aggregating the scores of the selected cities. To do the score aggregation we simply average the values of each feature of the selected cities.

Since the users are asked to express their preferences by selecting cities only from a subset of the dataset, we need to ensure that this selection shortlist is diverse enough to represent the cities in the dataset. To do this, we make use of the clustering results discussed in Section 3.3. For each cluster, we take top *k* characteristic cities of the cluster (the ones which are nearest to the centroid), randomly select two cities from this subset and finally add them to the selection shortlist[1]. Given that the clusters are diverse among each other, this procedure assures that the resulting selection shortlist will also contain diverse cities that represent the dataset. The remaining spots in the selection shortlist are filled with randomly selected cities from the dataset. This accounts for the size difference of the clusters. After a bit of experimentation, we found that 16 is an appropriate value for *k*. The randomness element in this method enables the system to have numerous, diverse but equivalent selection shortlists.

We limit the amount of cities the user can select from the shortlist, to a number from three to five. This helps ensuring that the system has enough data to work with for generating the initial user profile, and avoids having cases where users select all displayed cities, which end up in generic profiles with averaged out feature values.

In Section 4.3, we describe the user experience and interface aspects of preference elicitation in more details.

[1]Cluster 4 is an exception when applying this method as it is very small in size (only five instances). We select only one city from this cluster instead of two.

4.1.2. City Recommendation

Given that both the users and the cities are now similarly represented in the same vector space, we can apply a distance metric to calculate the distance a given user profile and the cities in the dataset. To accomplish this, we apply the Euclidean distance metric:

$$d(u,c) = \sqrt{\sum_{k=1}^{n}(u_k - c_k)^2} \tag{4.1}$$

where vector u represents the user profile and vector c represents a given city in the dataset. n is the number of dimensions (features), which is eight in our case, while u_k and c_k represent the *k-th* features of the user profile vector and the city vector, respectively. To recommend cities to a user with profile u, we calculate the distances between the user profile and every city in the dataset and finally recommend the cities which are the closest to the user profile with respect to the Euclidean distance, i.e., the cities which exhibit the lowest Euclidean distances from the user profile. Note that prior to calculating the Euclidean distances, we normalize the features by applying min-max scaling in order to avoid any bias from the difference in feature value ranges.

Table 4.1 shows the top recommendations for some example user profiles (feature values are scaled). Profile u^{Alice} is the result of aggregating the scores of Munich, Florence and Granada and has these u_k feature values: $u_1 = 0.35$ (*Average Temperature*), $u_2 = 0.45$ (*Cost of Living Index*), $u_3 = 0.52$ (*Food*), $u_4 = 0.31$ (*Arts and Entertainment*), $u_5 = 0.35$ (*Outdoors and Recreation*), $u_6 = 0.42$ (*Nightlife*), $u_7 = 0.27$ (*Average Precipitation*), $u_8 = 0.02$ (*Venue Count*). Profile u^{Bob} with feature values $0.56,\ 0.27,\ 0.60,\ 0.27,\ 0.38,\ 0.25,\ 0.32,\ 0.41$ is the result of aggregating the scores of Istanbul, Melbourne and Chiang Mai. While profile u^{Carol} with feature values $0.50, 0.56, 0.60, 0.24, 0.36, 0.30, 0.56, 0.28$ is the result of aggregating the scores of Singapore, Tokyo, Osaka and Toronto. After calculating the Euclidean distances, we observe that Verona, Marseille, Milan, Naples, Nuremberg are closest cities to profile u^{Alice}, Mexico City, São Paulo, Rome, Barcelona, Buenos Aires are closest to u^{Bob}, and Kobe, Sydney, Osaka, Kyoto, Hiroshima are closest to u^{Carol}.

	temp.	cost	food	art	outdoors	nightlife	precip.	venues	distance
u^{Alice}	0.35	0.45	0.52	0.31	0.35	0.42	0.27	0.02	
Verona	0.37	0.47	0.48	0.25	0.45	0.42	0.29	0.01	0.13
Marseille	0.41	0.42	0.48	0.33	0.46	0.35	0.21	0.01	0.16
Milan	0.36	0.52	0.55	0.38	0.31	0.35	0.37	0.08	0.18
Naples	0.46	0.42	0.59	0.25	0.34	0.32	0.33	0.01	0.19
Nuremberg	0.21	0.43	0.50	0.27	0.47	0.34	0.24	0.01	0.21
u^{Bob}	0.56	0.27	0.60	0.27	0.38	0.25	0.32	0.41	
Mexico City	0.47	0.09	0.63	0.26	0.25	0.36	0.23	0.60	0.34
São Paulo	0.57	0.20	0.46	0.48	0.37	0.38	0.50	0.48	0.35
Rome	0.46	0.47	0.61	0.29	0.33	0.28	0.29	0.09	0.39
Barcelona	0.49	0.39	0.57	0.27	0.27	0.45	0.22	0.11	0.41
Buenos Aires	0.50	0.14	0.56	0.45	0.26	0.35	0.38	0.10	0.43
u^{Carol}	0.50	0.56	0.60	0.24	0.36	0.30	0.56	0.28	
Kobe	0.46	0.58	0.67	0.20	0.26	0.29	0.52	0.04	0.28
Sydney	0.53	0.58	0.54	0.17	0.58	0.19	0.48	0.14	0.31
Osaka	0.47	0.53	0.71	0.19	0.11	0.43	0.54	0.14	0.33
Kyoto	0.45	0.58	0.60	0.45	0.27	0.25	0.64	0.07	0.33
Hiroshima	0.41	0.58	0.72	0.12	0.25	0.27	0.61	0.03	0.34

Table 4.1.: Top recommendations for some sample user profiles

4.1.3. Integrating Critiquing into the Recommendation Strategy and Refining User Profiles

Having defined the way how to infer the user profiles and how to recommend cities, we now have the foundation of the city recommender system. In this section, we explore how to improve the our recommendation strategy by employing a mechanism called *critiquing*, a common element in conversational recommender systems. As discussed in the literature review, critiquing is a helpful method to acquire user preferences and generally helps improving the efficiency of interaction with the system [20].

To give users more control on their preference profile, we ask the them to provide some feedback on an initial set of recommendations generated based on the cities picked from the selection shortlist, as described in Section 4.1.2. The requested feedback is in the form of a rating for the city features. For example, after the user is shown the temperatures of the initially recommended cities, he/she is asked if the temperature attribute should be lower, much lower, higher, much higher or is just right. The feedback question is repeated for all the other features, except for *Venue Count* and *Average Precipitation* as these are not very interpretable from the user's perspective.

Using this feedback, we update the user profile scores to attain a more refined preference model for the user.

Given that the feedback question is posed in the form of a five point rating scale, we create a mapping of the answers to concrete values, which quantify by how much should each feature value be increased or decreased. Taking advantage of the natural symmetry, we assign "should be much higher" answer to w, "should be higher" to $w/2$, "just right" to 0, "should be lower" to $-w/2$ and "should be much lower" to $-w$ where w quantifies the value of one unit of change. We assign w to a value from the interval $]0, 1[$. Exploiting the fact that the normalized features of the user profile take values in the segment $[0, 1]$ (as a result of applying min-max scaling), adding the values assigned to the answers (w, $w/2$, $0, -w/2$ and $-w$) to the feature value, allows us to update the initial user profile and reflect the user's feedback. If the newly updated feature value falls outside of the $[0, 1]$ segment, we simply reset it to 0 if it's a negative value or reset it to 1 if it's a value larger than 1.

The weight that the critiquing step has on the final user profile, depends on the value of w. A large w value correspond to a higher weight of the critiquing step, while a small w value corresponds to a lower weight. Table 4.2 shows the effect that the value of w has on the final recommendations. The user profile u^{Alice} is the result of aggregating the scores of Munich, Paris, and London and has these u_k feature values: $u_1 = 0.25$ (*Average Temperature*), $u_2 = 0.58$ (*Cost of Living Index*), $u_3 = 0.54$ (*Food*), $u_4 = 0.31$ (*Arts and Entertainment*), $u_5 = 0.34$ (*Outdoors and Recreation*), $u_6 = 0.39$ (*Nightlife*), $u_7 = 0.27$ (*Average Precipitation*), $u_8 = 0.13$ (*Venue Count*). While u^{Bob}, u^{Carol}, u^{Eve}, are the results of using u^{Alice} as the initial profile and applying a critiquing step which consists of this feedback: average temperature should be much higher (w), cost should be lower ($-w/2$) and the other aspects/scores are just right (0). The difference between u^{Bob}, u^{Carol} and u^{Eve} is the value of w used for updating the initial profile. u^{Bob} is the result of using $w = 0.3$, u^{Carol} the result of using $w = 0.4$, while u^{Eve} the result of using $w = 0.5$. After some experimentation, we chose to set $w = 0.4$ in the prototype, as it generally led to fairly balanced user profiles.

In the next Chapter 5, we evaluate whether this method leads to a better perceived quality of the recommendations compared to the plain recommendation strategy of Section 4.1.2, based on the results of a user study. Meanwhile, we continue this chapter by describing the design of the recommender system.

	temp.	cost	food	art	outdoors	nightlife	precip.	venues	distance
u^{Alice}	0.25	0.58	0.54	0.31	0.34	0.39	0.27	0.13	
London	0.29	0.60	0.53	0.26	0.35	0.44	0.23	0.20	0.12
Milan	0.36	0.52	0.55	0.38	0.31	0.35	0.37	0.08	0.20
Paris	0.30	0.64	0.65	0.33	0.17	0.35	0.23	0.15	0.23
Frankfurt	0.25	0.50	0.48	0.27	0.51	0.33	0.24	0.03	0.23
Luxembourg	0.21	0.66	0.54	0.15	0.43	0.41	0.31	~0.01	0.24
u^{Bob}	0.55	0.43	0.54	0.31	0.34	0.39	0.27	0.13	
Barcelona	0.49	0.39	0.57	0.27	0.27	0.45	0.22	0.11	0.14
Rome	0.46	0.47	0.61	0.29	0.33	0.28	0.29	0.09	0.17
Athens	0.55	0.37	0.56	0.35	0.31	0.36	0.14	0.04	0.18
Naples	0.46	0.42	0.59	0.25	0.34	0.32	0.33	0.01	0.19
Cagliari	0.48	0.40	0.59	0.33	0.29	0.34	0.15	~0.01	0.21
u^{Carol}	0.65	0.38	0.54	0.31	0.34	0.39	0.27	0.13	
Athens	0.55	0.37	0.56	0.35	0.31	0.36	0.14	0.04	0.19
Barcelona	0.49	0.39	0.57	0.27	0.27	0.45	0.22	0.11	0.20
Tel Aviv	0.63	0.56	0.48	0.39	0.39	0.39	0.20	0.03	0.24
Rome	0.46	0.47	0.61	0.29	0.33	0.28	0.29	0.09	0.25
Córdoba	0.54	0.32	0.50	0.31	0.29	0.55	0.22	0.01	0.25
u^{Eve}	0.75	0.33	0.54	0.31	0.34	0.39	0.27	0.13	
Athens	0.55	0.37	0.56	0.35	0.31	0.36	0.14	0.04	0.26
Durban	0.66	0.16	0.52	0.33	0.42	0.31	0.36	~0.01	0.28
Barcelona	0.49	0.39	0.57	0.27	0.27	0.45	0.22	0.11	0.29
Tel Aviv	0.63	0.56	0.48	0.39	0.39	0.39	0.20	0.03	0.30
Córdoba	0.54	0.32	0.50	0.31	0.29	0.55	0.22	0.01	0.30

Table 4.2.: Effect of w value on the recommendations

4.2. Recommender System Design

The goal of this section is to provide a high-level overview of the components of the recommender system and how they interact with each other.

The system architecture follows modularity principles to achieve a level of independence between the components. This helps facilitating the development of the prototype and accommodating for frequent incremental and iterative changes to the prototype. Figure 4.1 depicts a high-level abstraction of the system architecture.

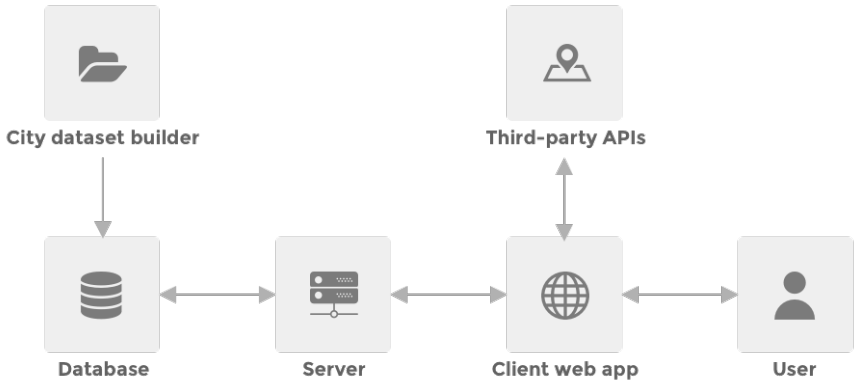


Figure 4.1.: System architecture

4.2.1. City Dataset Builder

The city dataset builder is the entity responsible for acquiring the climate, cost, and venue data about the cities and carrying out the city characterization process we described in Chapter 3. In other words, it executes a series of tasks related to data acquisition, data preparation, feature engineering, and data analytics in an orderly fashion, and finally saves the results in the database. For the purposes of our prototype, this entity is represented by a bundle of Python and R scripts which are run manually. As part of future work, an automated data pipeline could be employed without much effort. We continue the discussion about the data pipeline and its advantages in Chapter 6.

4.2.2. Server

This entity implements the recommendation strategies and exposes the necessary RESTful[2] API endpoints for clients (web, mobile or other) to serve city recommendations to the users. The server acts as a connection point between the entities of the system. The first endpoint it exposes accepts GET requests on path */api/cities*, and each time it is queried, it returns a list representative cities selected according to the method described in Section 4.1.1. The endpoint for recommendations accepts GET requests on path */api/recommendations*. The user preference information can be passed via url

[2]A widespread architectural style that is commonly used for developing web services.

parameters and the endpoint responds with an ordered list of city recommendations. If recommendation critiquing information is passed to the url parameters as well, the server uses the recommendation strategy with critiquing as described in Section 4.1.3, otherwise, it generates the recommendations using the plain recommendation strategy as described in Section 4.1.2.

4.2.3. Database

The database is used to save the city dataset into persistent storage. Given that the city dataset is static, the need for a separate database entity might seem redundant at first, as serving the city dataset from static files in the server would have worked just as well. However, that would introduce an undesirable coupling between the data and the implementation logic, and the server would have to be redeployed each time a record in the city dataset needs to be updated. Using a database allows for more involved queries and independent changes to the dataset. Furthermore, the database is used to persist some usage metrics of the prototype. We discuss these metrics in more detail in Chapter 5.

4.2.4. Client Web App

This entity is the only user facing part of the recommender system. It's goal is to provide a consistent user experience during the whole interaction of the user with the system. The client web application interacts with the server API to fetch city details and recommendations generated by the system, while being oblivious to the implementation details of the recommendation strategies. This entity also uses third-party APIs for displaying additional information about the cities, e.g., city pictures and static map images showing the location. As in any user-facing application, the interaction and interface design are important elements of the overall user experience, therefore, we aimed to adhere to design principles when defining the user interaction and interface design of the prototype application. We describe more details on the process in the next section.

4.3. User Interaction and System Interface

Before beginning with the implementation of the client web app, we started out by creating low fidelity paper mock-ups of the user interface, as shown in Figure 4.2. This helped us test various concepts for the user interaction and interface design, with low costs in terms of time and effort. After iterating on the paper mock-ups, we switched to higher fidelity digital mock-ups created using Sketch[3]. Primarily, we aimed to achieve a good balance between usability and aesthetics. The design decisions were eventually made based on some user feedback, previous experience and popular established principles in the field, like Nielsen's ten heuristics for usability [48].

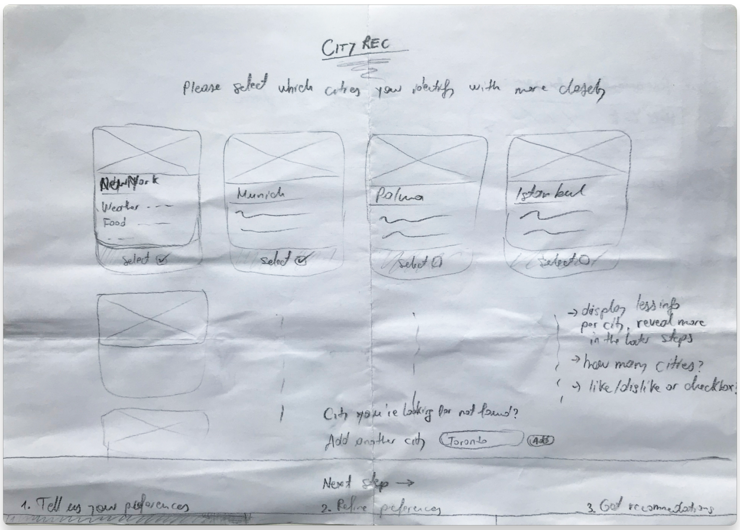


Figure 4.2.: Low fidelity paper mock-up

After doing a couple more iterations on the high fidelity digital mock-ups, we established the look and feel of the interface and focused on the development. Figure 4.3 shows the preference elicitation page. As discussed in Section 4.1.1, the system only needs a list of cities liked by the user to build the initial user preference profile. Therefore, this page shows a list of representative and the user is asked to choose the ones which best reflect his/her preferences. The call-to-action banner on the top is used to explain to the users what input the recommender system expects to receive. Considering that not all users might be accustomed to interacting with recommender

[3]A tool for designing user interfaces, websites, and icons; widely popular among designers. See https://www.sketchapp.com/

systems in general, we provide further information, via a non-intrusive tooltip that activates on mouse over or click. On the bottom of the page, the users are also provided with an option to refresh the city list in case they are not familiar with the presented cities.

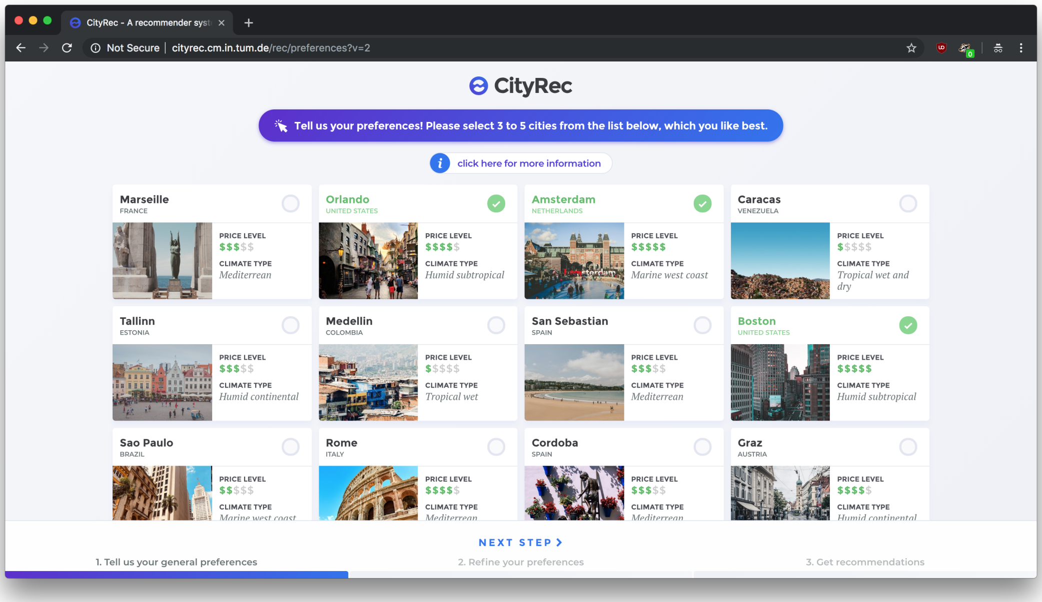


Figure 4.3.: Preference elicitation page

To achieve learnability and consistency across the interface, the rest of the pages follow similar structure. Figure 4.4 shows the recommendations critiquing page, where the user is presented with a set of four initial recommendations without specifying an order explicitly, and asked to rate various aspects of these recommendations. Based on this feedback the system refines the user profile and generates new recommendations, as shown in Figure 4.4. In the final recommendations page, the user is finally presented with a ranked list of cities which best fit his/her preference profile (Figure 4.5).

Across each page, a stepper element is shown on the bottom the screen, which informs the user about the progress and the state of the system.

At each step, the system gradually displays more information about the cities in order to avoid information overload. For example, in the first step (preference elicitation page) the system only displays the price level, climate type, country, and image of the cities. While in the last step (final recommendations page), the system additionally

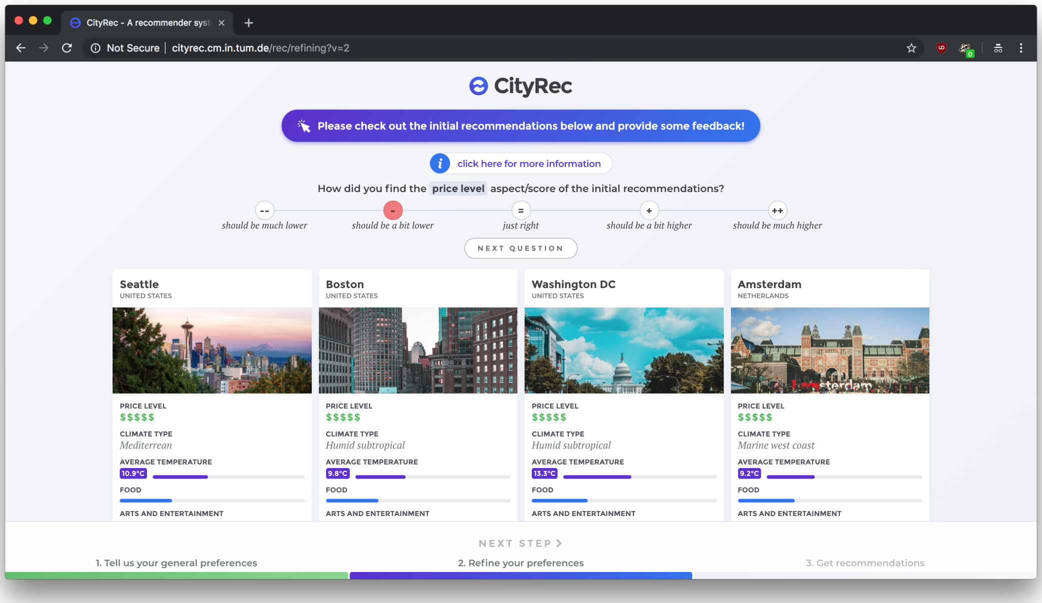


Figure 4.4.: Critiquing step

displays the location on the map and information about the other user-relevant city features available in the dataset.

4.3.1. Mapping City Features to Interface Elements

Since the city features in our city dataset are numerical, we needed to find a way to convey the information they present to the user, in an easy to interpret way. We renamed *Cost of Living Index* to "Price Level" as it is more interpretable from the user's perspective, and used equal-frequency binning to discretize this numerical feature into five bins; cities with lower cost of living index are assigned to a smaller bin number, while cities with higher cost of living index are assigned to a larger bin number. Using this information we display a rating indicator as shown in Figure 4.6.

For features *Arts and Entertainment*, *Food*, *Nightlife* and *Outdoors and Recreation* we show a bar chart so that users can easily compare the cities and identify the differences. For *Average Temperature* we also add the exact number, as it helps with the interpretation. These elements are as well shown in Figure 4.6.

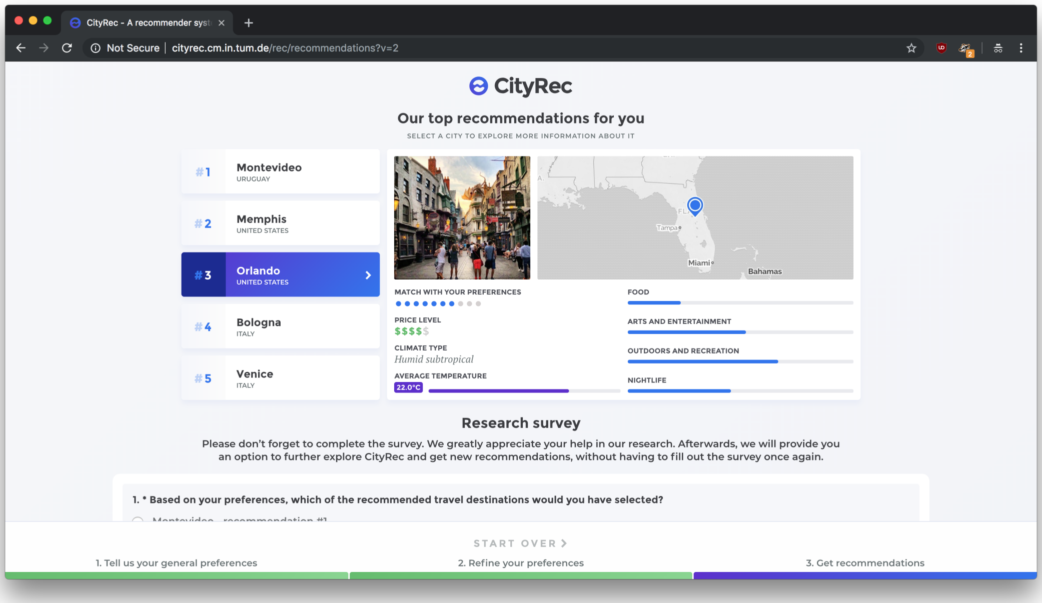


Figure 4.5.: Final recommendations page

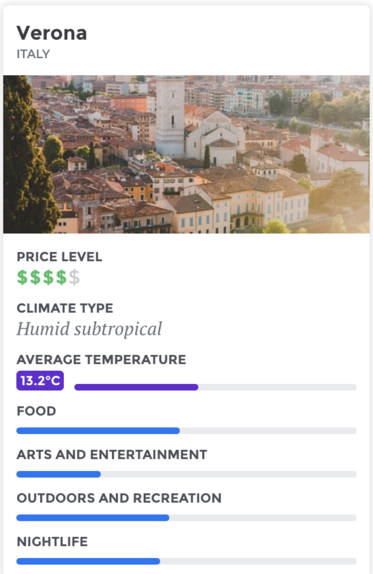
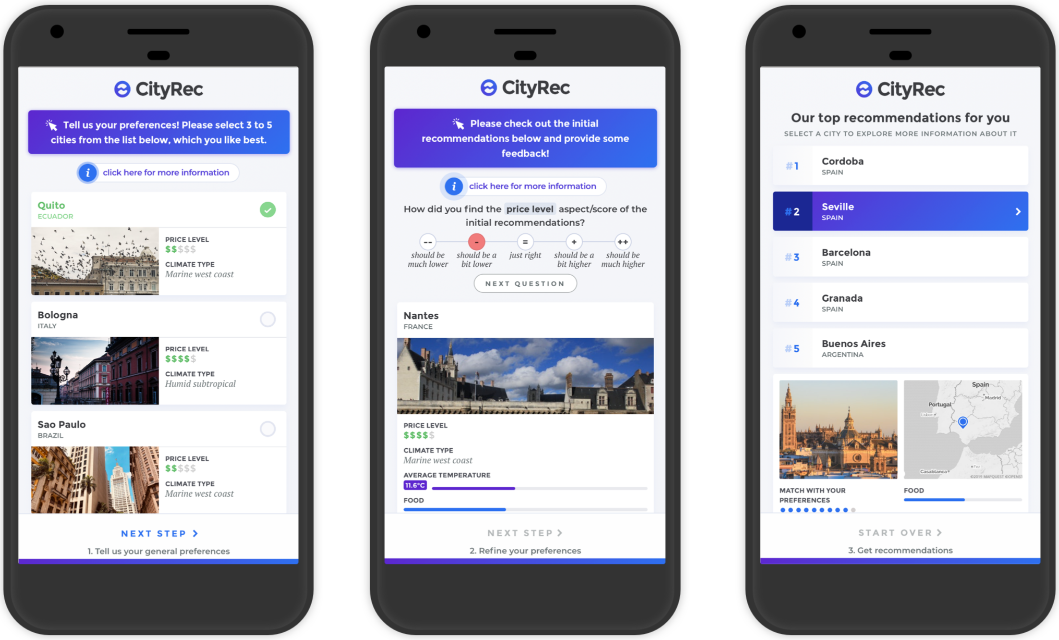


Figure 4.6.: City features overview

4.3.2. Responsive Design

Considering that the prototype will ultimately be used to conduct a online user study, we needed to make sure that the interface is usable in mobile devices as well. This was achieved mostly through the usage of CSS media queries, a useful CSS feature for building responsive interfaces[4]. Figure 4.7 depicts what the interface looks like on mobile devices.



(a) Preference elicitation (b) Critiquing step (c) Final recommendations

Figure 4.7.: System's interface on mobile

The user experience of our prototype on mobile exhibits certain drawbacks when compared to the experience on larger, more standard screen sizes. For example, the information displayed about the cities is not as easy to take in, as the user has to do a considerable amount of scrolling. However, the responsive design of the interface enables mobile device users to specify their preferences, receive recommendations, and achieve similar interaction flow as on devices with larger screens.

[4] https://developer.mozilla.org/en-US/docs/Web/CSS/Media_Queries/Using_media_queries

4.4. Implementation of the Prototype

In this section, we briefly go over the implementation details of the prototype. We used a relatively modern tech stack based on NodeJS, Express, MongoDB, and ReactJS for building the prototype application.

The server-side logic is implemented in Javascript. NodeJS, a Javascript runtime based on Chrome's V8 Javascript engine[5], makes it possible to run Javascript code in the server. Furthermore, the server makes use of Express — a popular web framework[6] — to serve the API endpoints and deal with routing.

The client web app is based on ReactJS[7], a very popular Javascript library maintained by Facebook. This library allows us to create reusable UI components and share code easily across different parts of the application.

This selection of server and client side technologies allows us to write the whole application code base in Javascript, and be able to apply iterative changes to the prototype without much effort.

We choose MongoDB[8] for database management as it allows for flexible structure of records and fits well with the rest of the technologies in our stack. Furthermore, the records in our database (cities and application usage documents) are self-contained and there are no dependencies between collections[9] in the database, making our use case a good fit for MongoDB.

Finally, for the tasks belonging to the city dataset builder entity in our system, we use Python and R. These languages are backed by large developer communities and have a rich collection of libraries for tasks related to data analytics.

[5]https://nodejs.org/en/
[6]https://expressjs.com/
[7]https://reactjs.org/
[8]https://www.mongodb.com/
[9]A collection is the equivalent of a table in SQL terms

5. Evaluation of the Prototype

In this chapter, we present and discuss the results of the user study we conducted for the evaluation of the CityRec prototype. Based on the gathered feedback and insights, we evaluate the perceived quality of the recommendations generated by the system and test hypotheses relevant to the research questions of this thesis. Initially, we discuss the structure of the user study and the adaptations applied to the prototype prior to using it in the study. We then focus on the hypotheses and the statistical analysis of the survey results. Finally, we discuss the findings of the user study and give an interpretation of the results.

5.1. Experimental Setup

Taking advantage of the fact that the CityRec prototype is based on a web application, we are able to conduct the user study entirely online. This enables us to reach a larger amount of participants, as the users can interact with the prototype and fill out the survey at the comfort of their own desktop or mobile browsers. To inform the participants what the user study is about, we added a landing page to the prototype web application which briefly describes the content of the study (Figure 5.1).

We also added a few mechanisms to keep track of some usage metrics, which help monitoring how the user interacts with the system, e.g., choices made at each step, time taken to specify the preferences, click count, device type and other similar metrics. Considering that we want to assess whether the critiquing recommendation strategy described in Chapter 4 leads to a better perceived quality of the recommendations compared to the plain recommendation strategy, we have to apply some further adaptations to the prototype. More specifically, we need to create two versions of our prototype system: one which generates recommendations using the plain recommendation strategy without the critiquing step, and another which generates recommendations using the critiquing recommendation strategy. Given that the two versions are otherwise identical, we can test the effect that this variation has on the perceived system quality.

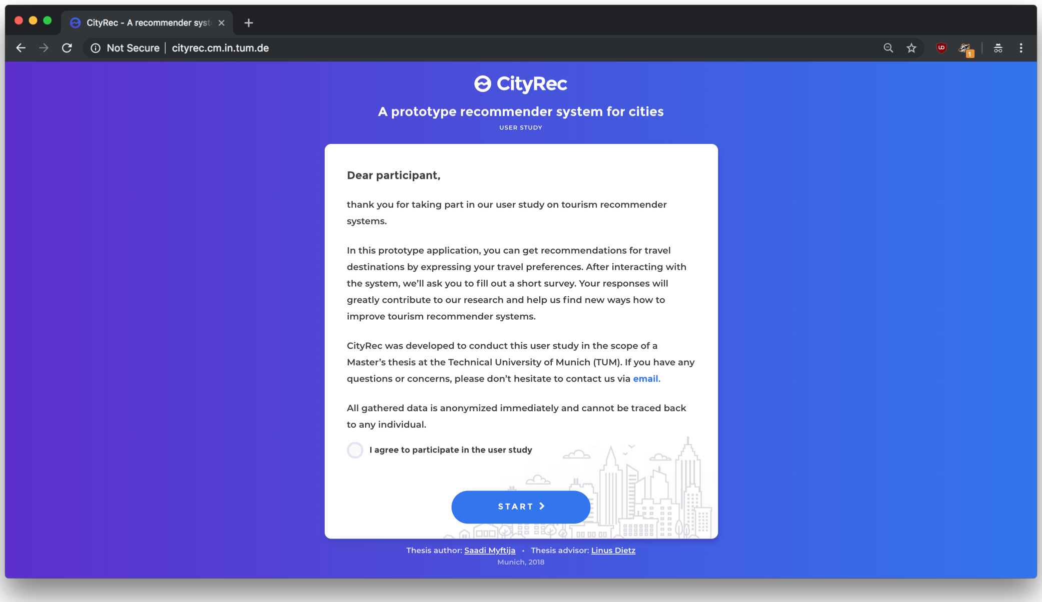


Figure 5.1.: Landing page

This procedure is often referred to as A/B testing. We continue the discussion on how we employed A/B testing in our evaluation approach in the next section.

5.1.1. A/B Testing

A/B testing is a method to compare two versions of a system through the use of randomized experiments, with the goal of determining which of the versions performs better with respect to certain performance indicators. By introducing a single variation between the two versions and keeping the rest identical, A/B tests enable investigating the effect that this variation has on the users' behavior. A/B testing and randomized experiments have become increasingly popular in the tech industry and are an essential part to innovation in many fields [4, 32].

In our prototype, the variation we are interested in is the critiquing step. This is the only independent variable of interest in our study. Therefore, we create an additional version of the prototype where the critiquing step is omitted from the system interaction and the recommendations are generated using the plain recommendation strategy. In the rest of this chapter, we refer to this version as the baseline version. Figure 5.2 shows the slight interface adaptations for the baseline recommender version. The critiquing recommender version remains as described in Chapter 4 and needs no further adaptations.

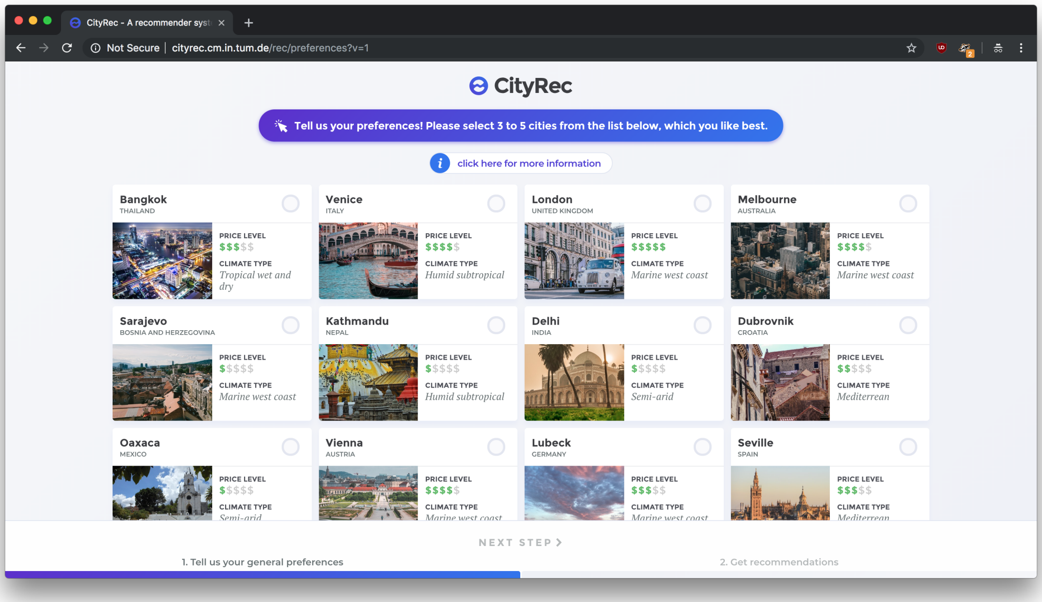


Figure 5.2.: Baseline recommender version

In order to avoid any bias when distributing the user study participants to the two versions, we randomly assign each participant to a version immediately after the interaction with the system starts at the landing page (Figure 5.1).

5.1.2. Survey Questions

Now that we have identified the independent variable of interest and defined the two versions to be compared, we focus on the survey part of the user study. The purpose of this survey is to gather feedback from the participants about their interaction with the system, and enable us to assess the perceived qualities of the recommender system such as usefulness, usability, satisfaction, and others. In the next section, we discuss the corresponding dependent variables that we measure in our study.

To determine the survey questions, we make use of the ResQue questionnaire, a validated, user-centric evaluation framework for recommender systems [50]. The questionnaire contains questions/statements such as: "The items recommended to me matched my interests.", which are answered on a 5-point Likert scale: *1:* Strongly disagree, *2:* Disagree, *3:* Neutral, *4:* Agree, *5:* Strongly agree. We used a subset of these questions, which were most relevant to our prototype and slightly adjusted some questions where necessary. We list the questions of this subset below:

- The travel destinations recommended to me by CityRec, matched my interests
- The recommender system helped me discover new travel destinations
- I understood why the travel destinations were recommended to me
- I found it easy to tell the system what my preferences are
- I found it easy to modify my taste profile in this recommender system
- The layout and labels of the recommender interface are adequate
- Overall, I am satisfied with this recommender system
- I would use this recommender system again, when looking for travel destinations

We add one more question to this set: "Based on your preferences, which of the recommended travel destinations would you have selected?" which can be answered by selecting one of the five recommendations or "None".

The set of questions we discussed previously helps us evaluate the perceived qualities of the recommender system. We are furthermore interested in gathering some information about the traveling habits and demographics of the participants. Hence, we add some questions about the age group, gender, travel frequency, and importance of some tourism aspects. These aspects are:

- Cost,
- Weather,
- A variety of cafes and restaurants,
- Plentiful cultural and entertainment attractions,
- A multitude of outdoor activities,
- Plentiful nightlife hotspots, and
- An abundance of shops and services.

The participants are asked to rate them on a 5-point scale: *1:* Not important at all, *2:* Slightly important, *3:* Important, *4:* Very important, *5:* Extremely important. The rated aspects can be naturally mapped to the city features we discussed in Chapter 3.

Lastly, the participants are also given the opportunity to write an optional comment regarding their experience with the system.

The survey page is shown after the interaction with the recommender system has finished and participants can answer the questions without having to leave the prototype web application. Once the participant submits the survey, the answers are bundled with the usage metrics of the interaction session and finally saved in the database.

5.1.3. Measured Dependent Variables

The main dependent variable that we are interested in measuring in this study is the *perceived accuracy of recommendations*. This variable corresponds to the survey question about how well the recommendations matched the user interests. By analyzing the responses to this question, we can measure the effect that our independent variable — *version* — has on the *perceived accuracy of recommendations*.

We also want to investigate whether *version* has an effect on how the user assesses the interface. Therefore, the other dependent variable that we measure is the *perceived interface adequacy*, which corresponds to the survey question about how adequate the interface was.

In order not to miss out on observing any other unanticipated effects of the *version* independent variable, we also keep track of the other dependent variables inferred from the remaining survey questions, such as novelty of recommendations, explainability of recommendations, ease of use, ease of profile modification, satisfaction, future use, and, position of selected recommendation. However, these are not the primary focus of this user study. Similarly, we also measure other dependent variables based on usage metrics, such as clicks count in the interaction session and time taken for the interaction. We discuss more details about these in Section 5.4. Other extraneous variables, such as gender, age and travel frequency, are irrelevant to the user study. The user study results show that there was no notable difference introduced by these extraneous variables.

5.2. Descriptive Statistics for the User Study

In this section we summarize some statistics for the results of user study. The aim of this section is to help the reader get a feel of the data gathered throughout the user study and create a basic understanding of its structure.

The participation levels in the user study were promising, with 67 total participants in a time frame of less than two weeks. Participants were recruited by sharing the user study in social media, and in groups of friends and colleagues. Using random assignment, 34 of the participants were placed in the baseline version group and 33 were placed in the critiquing version group, resulting in a balanced distribution of participants across the two versions. 30 of the participants are female and 37 are male, while the age demographics are mostly dominated by the 21-30 age group, with 46 of the participants. The 31-40 age group follows with 13 participants and lastly the -20 and 41-50 have 5 and 3 participants each. Figure 5.3 shows the distribution of travel frequencies among the survey participants.

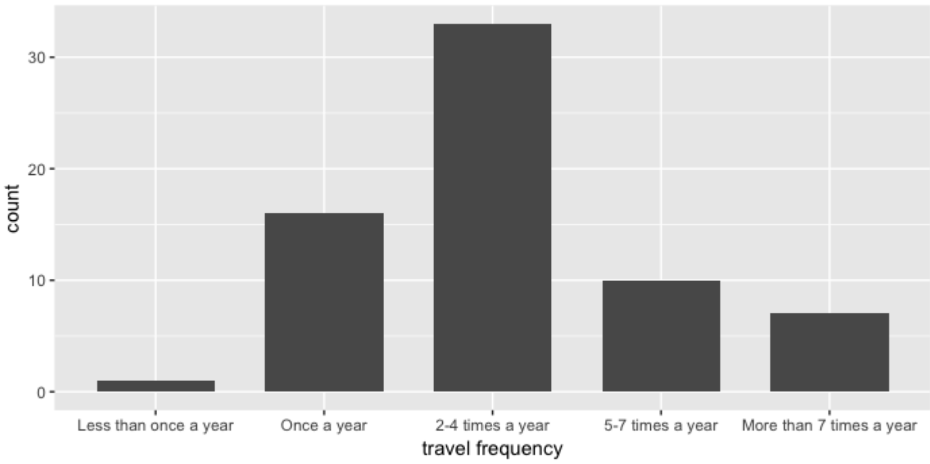
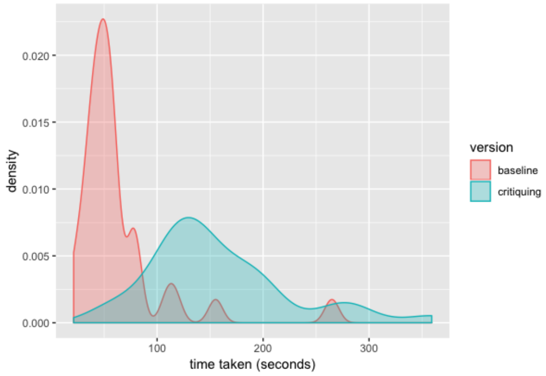
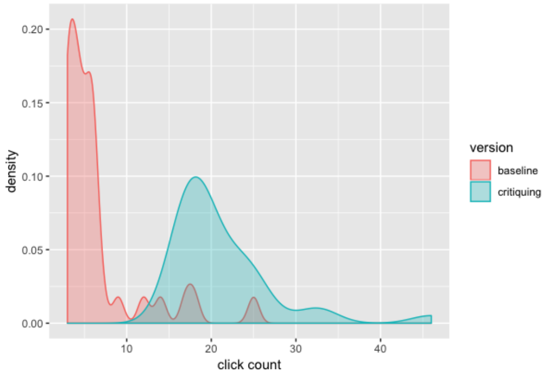


Figure 5.3.: Travel frequencies

47 of the participants interacted with the recommender system through a desktop browser, while the other 20 participants used a mobile browser. Further usage metrics that we kept track of are the click count and time taken during the interaction. Figure 5.4a shows the distribution of click counts, while Figure 5.4b shows the distribution of time taken for the interaction sessions.



(a) Distribution of interaction session durations



(b) Distribution of click counts

Figure 5.4.: Interaction metrics for baseline and critiquing versions

Figure 5.5 illustrates the answers of this survey question: "Based on your preferences, which of the recommended travel destinations would you have selected?". The participants could answer by selecting one of the recommended cities but also were given the option to select "None". The first ranked recommendation was the most frequently chosen one (17 times), followed by the third (16 times), second (14 times), fourth (10 times) and fifth (9 times). "None" was selected only once.

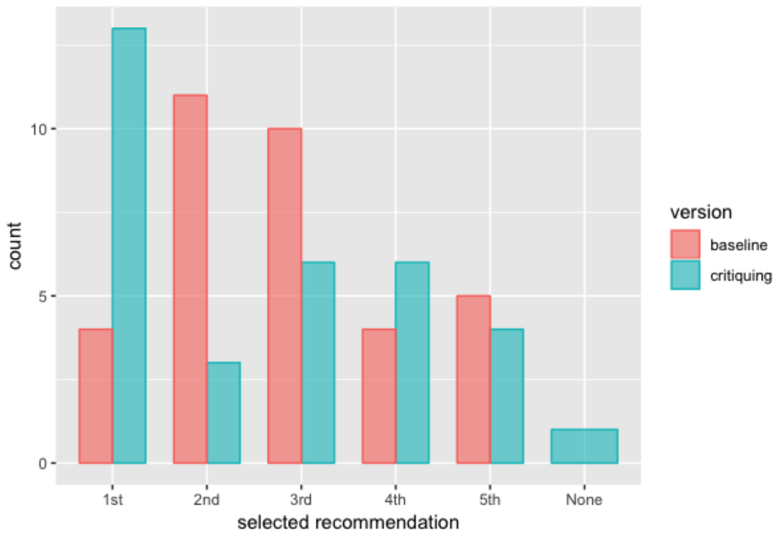


Figure 5.5.: Frequencies of selected recommendation positions

5.3. Main Hypotheses

In this section, we focus on formulating hypotheses based on our expectations for the outcomes of the user study. Having already defined the survey questions also helps in this process.

Our first main hypothesis revolves around the critiquing aspect of our proposed recommendation strategy:

Hypothesis 1 *Travel recommender systems which employ critiquing in their interaction style, can achieve higher perceived accuracy of recommendations compared to systems that use traditional methods of interaction.*

This hypothesis can be primarily associated with the survey question which asks the participants whether the shown recommendations matched their interests, corresponding to the *perceived accuracy of recommendations* dependent variable. We expect that the participants who interacted with the critiquing recommender version show a stronger level of agreement to the statement, compared to the participants shown the baseline recommender version.

The second hypothesis is about the interaction and interface aspect of the system. We expect that the critiquing recommender version and the baseline version attain similar levels of agreement in the question regarding the adequacy of the interface, which corresponds to the *perceived interface adequacy* dependent variable. To put it more formally:

Hypothesis 2 *Travel recommender systems which employ critiquing in their interaction style, can achieve the same level of perceived interface adequacy as systems that use traditional methods of interaction.*

5.3.1. Testing the Hypotheses

Initially, we look at the answers for the statement "The travel destinations recommended to me by CityRec, matched my interests" and investigate the differences in answers between the two groups. Figure 5.6 shows the distribution of the answers. For the baseline recommender version, 56% of the participants agreed or strongly agreed that the recommendations matched their interests, while for the critiquing recommender version, 85% of the participants agreed or strongly agreed that the recommendations matched their interests. This seems to support our main hypothesis, but we still need to test whether this difference between the two groups is statistically significant.

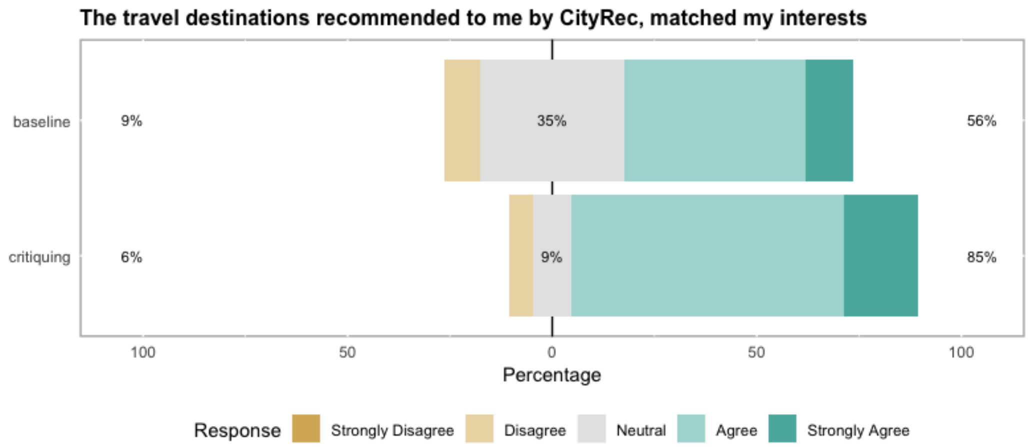


Figure 5.6.: Survey answers distribution for question: "The travel destinations recommended to me by CityRec, matched my interests"

In order for our analysis to lead us to the correct interpretation of the data, we need to make sure that an appropriate statistical test is selected for testing the hypotheses. The nature of the data is of fundamental importance when selecting a statistical test. Considering that the survey answers we are working with are based on the Likert scale, parametric tests are not appropriate, as they assume normality of the distribution and independence between the responses of the survey questions. Therefore we need to look into non-parametric procedures which do not rely on such assumptions. Literature suggests that the two-sample Mann–Whitney U test is suitable for Likert scale data and can be used to compare values for two groups [39]. Mann–Whitney U test is a special case of Kruskal-Wallis test. In the latter, multiple groups can be compared, while in the former, the number of groups to be compared is always two. The Likert scale data generated by the survey answers meets the criteria required by this statistical test:

- the dependent variable (*perceived accuracy of recommendations*) is ordinal,
- the independent variable (*version*) is nominal and has two categories (producing two groups),
- the observations between groups are independent as each participant is assigned one of the versions, therefore, there are no paired observations.

Based on this, we use the two-sample Mann–Whitney U test for testing if there is a statistically significant difference in the responses between the two groups. Considering that the distribution of the survey answers of the two groups are similar in shape and spread, the two-sample Mann–Whitney U test can be interpreted as a comparison of the medians of the two groups [39]. We formulate the null and alternative hypothesis for the two version groups, as follows:

H_0: The medians of perceived recommendation accuracy values for the two groups are equal ($m_{\text{critiquing}} = m_{\text{baseline}}$).

H_A: The median of perceived recommendation accuracy values for the critiquing version group is larger than the median for the baseline version group ($m_{\text{critiquing}} > m_{\text{baseline}}$),

where m denotes the population median. Performing the test with a significance level of 0.05, reports a p-value of 0.01547. We observe that $p < 0.05$, thus reject the null hypothesis H_0 and accept the alternative hypothesis H_A, concluding that the difference between the responses of the two groups is statistically significant. This result confirms **Hypothesis 1**, our first main hypothesis.

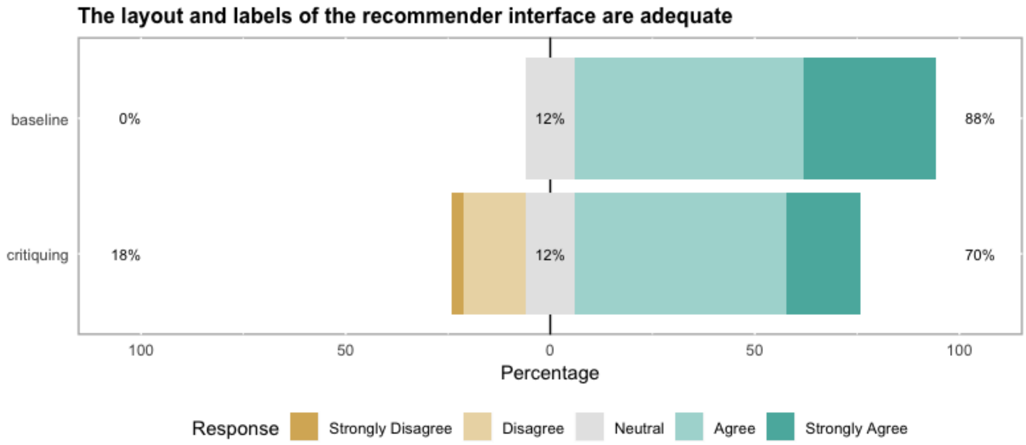


Figure 5.7.: Survey answers distribution for question: "The layout and labels of the recommender interface are adequate"

We now turn our attention to **Hypothesis 2** and analyze the answers for the statement "The layout and labels of the recommender interface are adequate". Figure 5.7 shows the distribution of the answers. For the baseline recommender version, 88% of the of the participants agreed or strongly agreed that the interface layout and elements are adequate, while for the critiquing recommender version 70% agreed or strongly agreed to the statement. Next, we test whether this difference is statistically significant. As in the previous case, we use the two-sample Mann-Whitney U test to perform this check. We formulate the null and alternative hypothesis for the two version groups as follows:

H_0: The medians of perceived interface adequacy values for the two groups are equal ($m_{\text{critiquing}} = m_{\text{baseline}}$).

H_A: The medians of perceived interface adequacy values for the two groups are not equal ($m_{\text{critiquing}} \neq m_{\text{baseline}}$),

where m denotes the population median. In this case, the claim of the hypothesis we are testing, **Hypothesis 2**, corresponds to H_0. Although this is not very usual, our choice of H_0 and H_A is still adequate as the null hypothesis H_0 should reflect a statement of no change between the groups [60]. Performing the Mann-Whitney U test with a significance level of 0.05, reports a p-value of 0.03602. We observe that $p < 0.05$, thus reject the null hypothesis H_0 and accept the alternative hypothesis H_A, concluding that the difference between the responses of the two groups is statistically significant. This result does not support **Hypothesis 2**.

5.3.2. Discussion

As shown in the statistical analysis of the previous section, the results of the user study confirm our **Hypothesis 1**. We were able prove that the critiquing recommender version achieved significantly higher perceived accuracy of recommendations compared to the baseline version, thus answering our research question **RQ2**. While accuracy is arguably one of the most important aspects of a recommender system, we are naturally interested to know whether this improvement in the perceived accuracy comes at a cost. As depicted in Figure 5.7, the perceived interface adequacy of the critiquing recommender version is significantly lower compared to the baseline version. This is not too surprising when considering the fact that the average time taken for interacting with the critiquing recommender was almost three times as large as the average time taken for interacting with the baseline version (Figure 5.4). Similarly, the average clicks count was also approximately three times higher for the critiquing version, compared to the baseline version. Another reason for the lower perceived interface adequacy of the critiquing recommender version might have been a possibly inappropriate choice of the interface elements for obtaining the ratings during the critiquing step. However, when asked how easy it is to tell the recommender system their preferences, the participants in both groups showed similar levels of agreement. As shown in Figure 5.8, for the baseline version 76% of the participants agreed or strongly agreed, while for the critiquing version 73% of the participants agreed or strongly agreed that it was easy to tell the recommender system their preferences. These insights bring to light an important challenge faced when designing travel recommender systems, which lies in finding a good balance in the trade-off between perceived recommendation accuracy and perceived ease of use of the system.

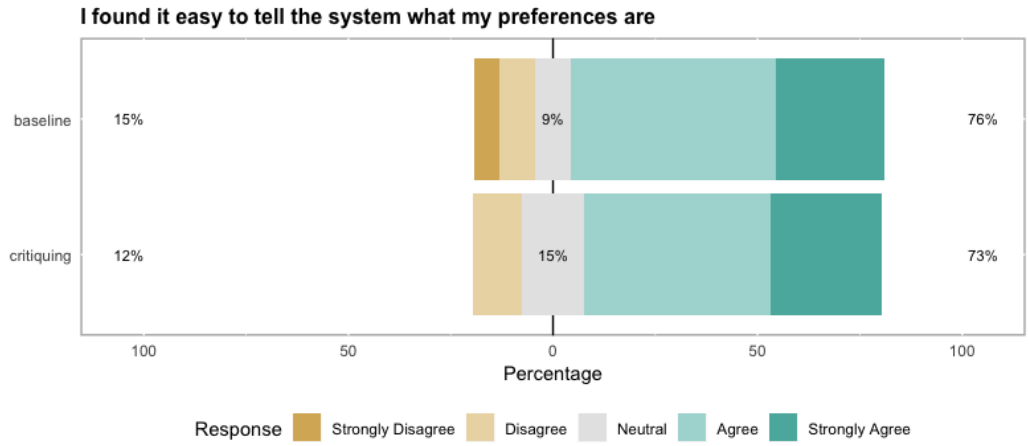


Figure 5.8.: Survey answers distribution for question: "I found it easy to tell the system what my preferences are"

On the other hand, the discrepancy between the levels of agreement for the perceived recommendation accuracy and the perceived interface adequacy for the critiquing recommender version, serves as evidence that the observed effect of significantly improved perceived accuracy was not simply caused by an interface improvement of the critiquing version.

In general, the rest of the survey answers were also quite supportive. 65% of the participants in the baseline version group and 73% of the participants in the critiquing version group, agreed or strongly agreed that the recommender system helped them discover new travel destinations. 56% of the participants in the baseline version group and 70% of the participants in the critiquing version group, agreed or strongly agreed that they understood why the travel destinations were recommended to them. Finally, 71% of the participants in the baseline version group and 76% of the participants in the critiquing version group, agreed or strongly agreed that they were overall satisfied with the recommender system. Besides helping us to test our hypotheses, the user study results served as a further confirmation that the city characterization method described in Chapter 3 can be effectively utilized in a travel recommender system.

5.4. Further Findings

Using the data gathered during the user study, we can reveal more insights about our prototype recommender system. Some of these insights are rather subtle, while some others more evident.

The survey questions we defined in Section 5.1.2 also include questions which ask the participants to rate the importance of some tourism aspects such as cost, weather, plentiful cultural and entertainment attractions, a multitude of outdoor activities and other aspects. As we mentioned previously, these aspects can be naturally mapped to the city features we discussed in Chapter 3. Therefore, we can consider the combination of these ratings as a self-evaluated user profile of each participant and see how it compares to the user profile inferred by the recommender system. Table 5.1 shows the correlations between the self-evaluated scores and the actual system-generated scores of the user profile features, grouped by version. Some aspects, such as cost and weather, are not included in this comparison as the survey questions for these aspects did not specify whether the participant prefers high or low scores in these aspects, but rather simply asked if the aspects themselves are important or not.

	Food	Arts and Entertainment	Outdoors and Recreation	Nightlife	Average correlation
Baseline	-0.04	0.01	-0.11	0.19	0.01
Critiquing	-0.01	0.34	0.45	0.59	0.34

Table 5.1.: Correlations between the self-evaluated scores and the system-generated scores of the user profile features

As shown in Table 5.1, the average correlation between the self-evaluated scores and the system-generated scores is considerably higher for the critiquing version group. This can be interpreted as an indication that the critiquing recommender version does a better job at capturing the intent and preferences of the user. Thus, this supports **Hypothesis 1** not being an artifact of longer interaction with the system.

Another interesting insight to explore is the distribution of the user profile scores among the participants of the user study. The distributions along each feature are shown in Figure 5.9.

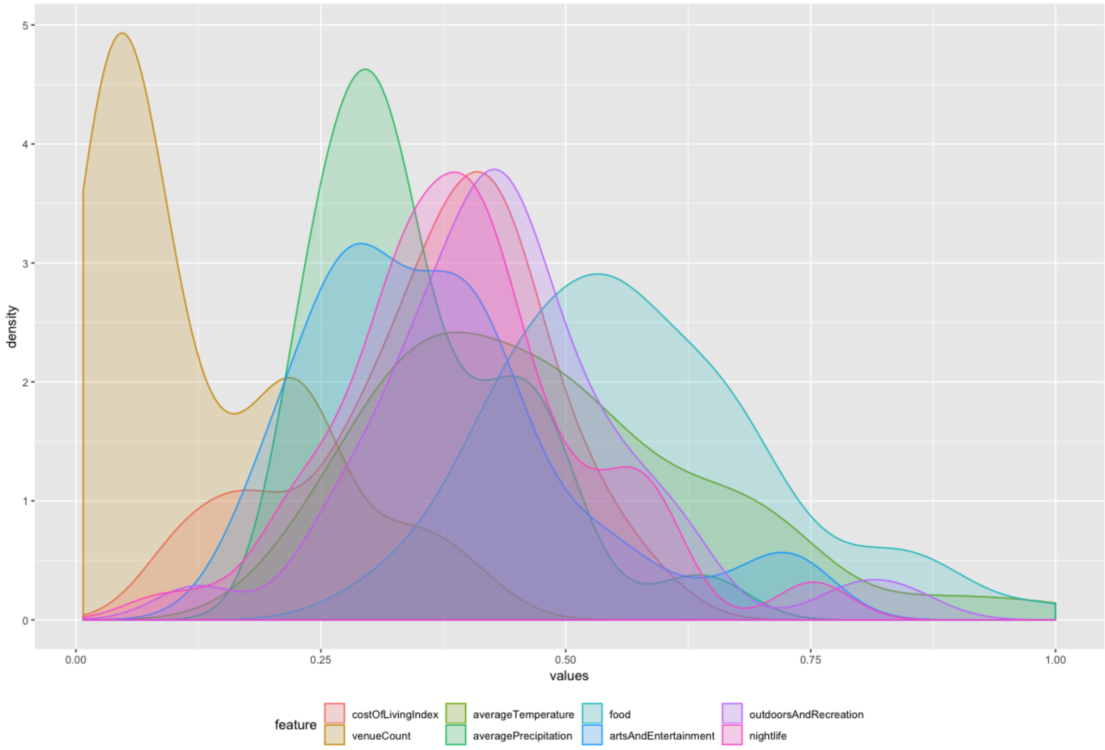


Figure 5.9.: Distributions of user profile scores

For the baseline version group, the average user profile u^{baseline} inferred by aggregating the user profile scores of the group's participants has these (scaled) feature scores: $u_1 = 0.43$ (*Average Temperature*), $u_2 = 0.40$ (*Cost of Living Index*), $u_3 = 0.50$ (*Food*), $u_4 = 0.31$ (*Arts and Entertainment*), $u_5 = 0.41$ (*Outdoors and Recreation*), $u_6 = 0.38$ (*Nightlife*), $u_7 = 0.33$ (*Average Precipitation*), $u_8 = 0.12$ (*Venue Count*). While for the critiquing version group, the average user profile $u^{\mathsf{critiquing}}$ has these (scaled) feature scores: $u_1 = 0.53$, $u_2 = 0.33$, $u_3 = 0.65$, $u_4 = 0.44$, $u_5 = 0.46$, $u_6 = 0.41$, $u_7 = 0.37$, $u_8 = 0.13$. These two average profiles are visually depicted in Figure 5.10. The system's top ten city recommendations for u^{baseline} would be: Verona, Marseille, Naples, Bologna, Rome, Milan, Barcelona, Florence, Turin, and Belfast. While for $u^{\mathsf{critiquing}}$ the top ten recommendations would be: Johannesburg, Naples, Rome, Buenos Aires, Durban, Barcelona, Athens, Milan, Marseille, and Bologna.

Next, we look at some statistics related to the diversity of the recommendations. Out of approximately 200 cities of the dataset, 149 were shown at least once in the the preferences page, at the start of the interaction with recommender system. 83 of these

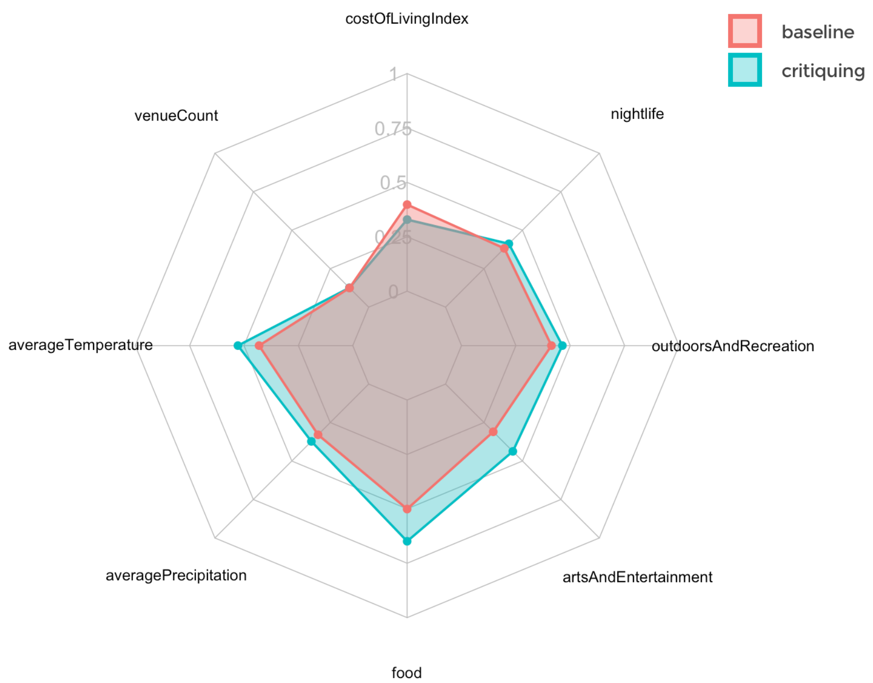


Figure 5.10.: Average user profile

cities were selected at least once by the participants. The most frequently selected cities were: Lisbon (10 times), Rome (9), Amsterdam (8), São Paulo (7), Rio De Janeiro (7), Punta Cana (7), Barcelona (7), Nice (6), Madrid (6), and London (6). The dominance of European cities in this list could possibly be explainable by the fact that most of the participants come from European countries.

Lastly, 86 of the cities were recommended at least once to the participants across all interaction sessions. The most frequently recommended cities were: Verona (18), Marseille (17), Naples (15), Rome (13), Bologna (12), Barcelona (12), Milan (11), Düsseldorf (11), Cologne (11), and Durban (10). As this metric is in a way an indicator of the recommendations diversity, it is useful to look at the difference between the two versions. In the baseline version, a total of 51 cities were recommended at least once to the participants, while in the critiquing version a total of 72 cities were recommended at least once across the sessions. This considerable difference could be interpreted as a suggestion that the critiquing version performs better compared to the baseline version, with respect to the diversity of recommendations.

With this final insight, we wrap up our discussion for the user study.

6. Conclusions and Future Work

In this thesis, we introduced an approach to characterize travel destinations based on inferring information from single POIs. After applying this method to characterize approximately 200 cities around the world, we built a conversational travel recommender system for cities. To answer our research questions, we used this recommender system to conduct a user study and investigated whether integrating critiquing into the interaction flow improves the perceived quality of travel destination recommendations. Throughout the chapters of this thesis we touched upon several facets of recommender systems; in the following, we mention some noteworthy aspects and summarize the conclusions of our work.

We started this thesis by reviewing essential concepts of recommender systems, such as recommendation techniques and user modeling. We discussed related work done in the area of travel recommender systems and identified the apparent absence of systems which recommend cities as destinations. Furthermore, we pointed out that a lot of existing approaches for destination characterization heavily rely on expert knowledge, thus hindering the scalability of the systems. This helped us to further motivate our research.

With the newly acquired insights in the literature review, we presented our solution to the city characterization task: a scalable approach to characterize destination cities, which relies on inferring information from single POIs. We integrated additional data sources in this technique with the goal of enriching the characterization of the cities by including climate and cost data. Furthermore, we described how we used cluster analysis to distinguish structure among the cities and validate the results of our proposed characterization approach. The main challenge in this process revolved around the extensive experimentation with feature selection and feature engineering that was necessary to find good and interpretable clusters. Another challenge we faced was the restricting rate limit for API requests imposed by Foursquare during venue collection.

Using the characterized cities dataset, we built a prototype travel recommender system. The challenge in this part was in finding the appropriate method to elicit the initial user preferences in a way that enriches the experience with the system. To model the user preferences, the system makes use of a non-intrusive method, which asks the user to select cities that they like from a representative city list. This representative list of cities was inferred from the clustering results. We further augmented the preference elicitation model by integrating a critiquing mechanism and adapting a conversational style interaction. Ultimately, we conducted a user study to investigate whether this augmentation leads to higher perceived quality of recommendations.

For conducting the user study, we employed A/B testing to compare the critiquing recommender version against a baseline version and determined a set of survey questions to investigate the perceived quality of the recommender system. We described in detail how we evaluated the recommender system using the user study results and augmented our choice for the used statistical methods. To address the challenging task of determining the assessment criteria, we made use of the ResQue evaluation framework for recommender systems. The results of the study were very supportive. We were able to prove that the critiquing recommender version achieved significantly higher perceived accuracy of recommendations compared to the baseline version, thus, confirming our hypothesis and answering our research question. Besides this, the study results served as a further confirmation that our proposed city characterization approach can be effectively utilized in a travel recommender system.

The user study also helped us uncover certain limitations of the system, such as recommendation diversity, and made it simpler for us to identify potential directions of future work. We treat this topic in the remainder of this chapter.

Future work

As part of future research, the travel recommender system and approaches presented in this thesis could be extended in numerous interesting ways. In our prototype recommender system, we did not focus on the diversity of the recommended cities. It would be interesting to employ mechanisms which introduce an element of diversity in the final recommendations, and investigate how that affects the perceived quality of recommendations.

Another interesting direction for future work would be to explore potential ways of extending our proposed destination characterization approach. Additional data sources could be analyzed with goal of inferring new engineered features to describe the cities, e.g., number of rivers and lakes, number of mountains, being coastal or continental, number of days with sunshine. Text mining methods could be employed to scan for even more data about the cities. It would also be interesting to study how to evaluate cities from a qualitative perspective, as the approach we used relies on quantitative aspects such as venue counts, which might be a limiting factor. Furthermore, it would be intriguing to inspect how these augmentations would affect the results of the cluster analysis.

The destination characterization approach could further be used for characterizing destinations of different granularity levels like regions or countries. It would be interesting to test the applicability of the our proposed approach in those scenarios and see what challenges emerge.

Some level of automation could be introduced in a useful way to the recommender system. The city dataset builder entity of the prototype — responsible for implementing the city characterization approach — could be adapted to employ an automated data pipeline. Besides making the implementation more elegant, this would allow to effortlessly characterize a given set of cities on demand. It would be interesting to run this data pipeline at different points in time and observe how the cities evolve with time.

Lastly, future work could also focus on the user interaction with the recommender system. Different interaction flows and interface layouts could be tried out. As previously discussed, there is a fundamental trade-off in recommender systems between perceived accuracy of recommendations and ease of use of the system, hence it would be interesting to investigate the effect that changes in interaction and interface have in the perceived accuracy and ease of use of the system.

Appendices

A. List of Selected Cities

In this appendix, we provide the list of the cities we selected for our travel destination characterization approach. The selection was based on tourist guides and online searches for popular tourist destinations. The cities marked with an asterisk (*) were discarded prior to starting the characterization process as they had too few Foursquare venues to work with.

Accra, Ghana
Aksum, Ethiopia *
Amsterdam, Netherlands
Antalya, Turkey
Antigua Guatemala, Guatemala *
Aspen, United States *
Asuncion, Paraguay
Athens, Greece
Auckland, New Zealand
Bangkok, Thailand
Barcelona, Spain
Beijing, China
Belfast, United Kingdom
Berlin, Germany
Bern, Switzerland
Bilbao, Spain
Bogota, Colombia
Bol, Croatia *
Bologna, Italy
Boston, United States
Bridgetown, Barbados *
Brussels, Belgium
Bucharest, Romania
Budapest, Hungary
Buenos Aires, Argentina
Busan, South Korea
Cagliari, Italy
Cairo, Egypt
Calgary, Canada
Cancun, Mexico
Cape Town, South Africa
Caracas, Venezuela
Cardiff, United Kingdom
Cartagena, Colombia
Casablanca, Morocco
Chamonix, France *
Charleston, United States
Chiang Mai, Thailand
Chicago, United States
Cologne, Germany
Colonia del Sacramento, Uruguay *
Copenhagen, Denmark

Cordoba, Spain
Cuzco, Peru
Davos, Switzerland *
Delhi, India
Denpasar, Indonesia
Dubai, United Arab Emirates
Dublin, Ireland
Dubrovnik, Croatia
Durban, South Africa
Düsseldorf, Germany
Edinburgh, United Kingdom
Fes, Morocco *
Florence, Italy
Fortaleza, Brazil
Frankfurt, Germany
Garmisch-Partenkirchen, Germany *
Genoa, Italy
Glasgow, United Kingdom
Gothenburg, Sweden
Granada, Spain
Graz, Austria
Guangzhou, China
Hamburg, Germany
Hanoi, Vietnam
Heidelberg, Germany
Helsinki, Finland
Hiroshima, Japan
Ho Chi Minh City, Vietnam
Hoi An, Vietnam *
Hong Kong, Hong Kong
Honolulu, United States
Hue, Vietnam
Hvar, Croatia *
Innsbruck, Austria
Ise, Japan *
Istanbul, Turkey
Jakarta, Indonesia
Jerusalem, Israel
Johannesburg, South Africa
Kampala, Uganda
Kampot, Cambodia *
Kathmandu, Nepal
Kobe, Japan
Krakow, Poland
Kuala Lumpur, Malaysia
Kuta, Indonesia
Kyoto, Japan
La Paz, Bolivia
Lagos, Nigeria
Las Vegas, United States
Leipzig, Germany
Lima, Peru
Lisbon, Portugal
Ljubljana, Slovenia
London, United Kingdom
Los Angeles, United States
Luang Prabang, Laos *
Lubeck, Germany
Luxembourg City, Luxembourg
Macau, China
Madrid, Spain
Mandalay, Myanmar *
Manila, Philippines
Marrakesh, Morocco
Marseille, France
Mecca, Saudi Arabia

Medellin, Colombia
Melbourne, Australia
Memphis, United States
Mexico City, Mexico
Miami, United States
Milan, Italy
Minneapolis, United States
Monaco, Monaco *
Mont-Tremblant, Canada *
Montevideo, Uruguay
Montreal, Canada
Moscow, Russia
Mumbai, India
Munich, Germany
Nairobi, Kenya
Nantes, France
Naples, Italy
Nara, Japan
New Orleans, United States
New York City, United States
Nice, France
Nuremberg, Germany
Oaxaca, Mexico
Okinawa, Japan
Olinda, Brazil
Orlando, United States
Osaka, Japan
Oslo, Norway
Palma de Mallorca, Spain
Panama City, Panama *
Paris, France
Pattaya, Thailand
Penang, Malaysia
Perth, Australia
Phnom Penh, Cambodia
Phuket City, Thailand
Pittsburgh, United States
Pokhara, Nepal *
Porto, Portugal
Potsdam, Germany
Prague, Czech Republic
Puerto Vallarta, Mexico
Punta Cana, Dominican Republic
Quebec City, Canada
Quito, Ecuador
Reykjavik, Iceland
Riga, Latvia
Rio De Janeiro, Brazil
Rome, Italy
Rossland, Canada *
Rotterdam, Netherlands
Saint Petersburg, Russia
Salvador Da Bahia, Brazil
Salzburg, Austria
San Diego, United States
San Francisco, United States
San Jose, United States
San Juan, Puerto Rico
San Sebastian, Spain
Santa Fe, United States
Santiago, Chile
Sao Paulo, Brazil
Sarajevo, Bosnia And Herzegovina
Seattle, United States
Seoul, South Korea
Seville, Spain

Shanghai, China
Shenzhen, China
Siem Reap, Cambodia
Siena, Italy *
Singapore, Singapore
Sofia, Bulgaria
Stockholm, Sweden
Sydney, Australia
Taipei, Taiwan
Tallinn, Estonia
Tel Aviv, Israel
Thimphu, Bhutan *
Tokyo, Japan
Toledo, Spain
Toronto, Canada
Tulum, Mexico *
Tunis, Tunisia
Turin, Italy
Ubud, Indonesia
Udaipur, India *
Valletta, Malta *
Vancouver, Canada
Varanasi, India *
Venice, Italy
Verona, Italy
Victoria, Canada
Vienna, Austria
Vilnius, Lithuania
Wailea-Makena, United States *
Warsaw, Poland
Washington DC, United States
Wroclaw, Poland
York, United Kingdom
Zagreb, Croatia

List of Figures

List of Tables

Bibliography

[1] L. Ardissono, A. Goy, G. Petrone, M. Segnan, and P. Torasso. "Intrigue: Personalized recommendation of tourist attractions for desktop and hand held devices." In: *Applied Artificial Intelligence* 17.8-9 (2003), pp. 687–714. DOI: 10.1080/713827254. eprint: https://doi.org/10.1080/713827254.

[2] L. Ardissono, A. Goy, G. Petrone, M. Segnan, and P. Torasso. "Ubiquitous User Assistance in a Tourist Information Server." In: *Adaptive Hypermedia and Adaptive Web-Based Systems*. Ed. by P. De Bra, P. Brusilovsky, and R. Conejo. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 14–23. ISBN: 978-3-540-47952-9.

[3] O. Averjanova, F. Ricci, and Q. N. Nguyen. "Map-Based Interaction with a Conversational Mobile Recommender System." In: *2008 The Second International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*. Sept. 2008, pp. 212–218. DOI: 10.1109/UBICOMM.2008.16.

[4] E. M. Azevedo, D. Alex, J. Montiel Olea, J. M. Rao, and E. G. Weyl. "A/B testing." In: (2018).

[5] R. Barta, C. Feilmayr, B. Pröll, C. Grün, and H. Werthner. "Covering the Semantic Space of Tourism: An Approach Based on Modularized Ontologies." In: *Proceedings of the 1st Workshop on Context, Information and Ontologies*. CIAO '09. Heraklion, Greece: ACM, 2009, 1:1–1:8. ISBN: 978-1-60558-528-4. DOI: 10.1145/1552262.1552263.

[6] M. Batet, A. Moreno, D. Sánchez, D. Isern, and A. Valls. "Turist@: Agent-based personalised recommendation of tourist activities." In: *Expert Systems with Applications* 39.8 (2012), pp. 7319–7329. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2012.01.086.

[7] D. Bawden and L. Robinson. "The dark side of information: overload, anxiety and other paradoxes and pathologies." In: *Journal of Information Science* 35.2 (2009), pp. 180–191. DOI: 10.1177/0165551508095781. eprint: https://doi.org/10.1177/0165551508095781.

[8] M. Ben Messaoud, I. Jenhani, E. Garci, and T. De Pessemier. "SemCoTrip: A Variety-Seeking Model for Recommending Travel Activities in a Composite Trip." In: *Advances in Artificial Intelligence: From Theory to Practice*. Ed. by S. Benferhat, K. Tabia, and M. Ali. Cham: Springer International Publishing, 2017, pp. 345–355. ISBN: 978-3-319-60042-0.

[9] R. Bidart, A. C. M. Pereira, J. M. Almeida, and A. Lacerda. "Where Should I Go? City Recommendation Based on User Communities." In: *2014 9th Latin American Web Congress*. Oct. 2014, pp. 50–58. DOI: 10.1109/LAWeb.2014.15.

[10] J. Borràs, J. de la Flor, Y. Pérez, A. Moreno, A. Valls, D. Isern, A. Orellana, A. Russo, and S. Anton-Clavé. "SigTur/E-Destination: A System for the Management of Complex Tourist Regions." In: *Information and Communication Technologies in Tourism 2011*. Ed. by R. Law, M. Fuchs, and F. Ricci. Vienna: Springer Vienna, 2011, pp. 39–50. ISBN: 978-3-7091-0503-0.

[11] J. Borràs, A. Moreno, and A. Valls. "Intelligent tourism recommender systems: A survey." In: *Expert Systems with Applications* 41.16 (Nov. 2014), pp. 7370–7389. ISSN: 0957-4174. DOI: 10.1016/j.eswa.2014.06.007.

[12] J. S. Breese, D. Heckerman, and C. Kadie. "Empirical Analysis of Predictive Algorithms for Collaborative Filtering." In: *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*. UAI'98. Madison, Wisconsin: Morgan Kaufmann Publishers Inc., 1998, pp. 43–52. ISBN: 1-55860-555-X.

[13] J. J. Burg, J. Ainsworth, B. Casto, and S.-D. Lang. "Experiments with the "oregon trail knapsack problem"." In: *Electronic Notes in Discrete Mathematics* 1 (1999), pp. 26–35.

[14] R. Burke. "Hybrid Recommender Systems: Survey and Experiments." In: *User Modeling and User-Adapted Interaction* 12.4 (Nov. 2002), pp. 331–370. ISSN: 1573-1391. DOI: 10.1023/A:1021240730564.

[15] R. Burke. "Hybrid Web Recommender Systems." In: *The Adaptive Web: Methods and Strategies of Web Personalization*. Ed. by P. Brusilovsky, A. Kobsa, and W. Nejdl. Berlin, Heidelberg: Springer, 2007, pp. 377–408. ISBN: 978-3-540-72079-9. DOI: 10.1007/978-3-540-72079-9_12.

[16] R. Burke. "Interactive Critiquing forCatalog Navigation in E-Commerce." In: *Artificial Intelligence Review* 18.3 (Dec. 2002), pp. 245–267. ISSN: 1573-7462. DOI: 10.1023/A:1020701617138.

[17] R. Burke. "Knowledge-based recommender systems." In: 2000.

[18] R. Burke and M. Ramezani. "Matching Recommendation Technologies and Domains." In: *Recommender Systems Handbook*. Ed. by F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor. Boston, MA: Springer US, 2011, pp. 367–386. ISBN: 978-0-387-85820-3. DOI: 10.1007/978-0-387-85820-3_11.

[19] P. Castells, N. J. Hurley, and S. Vargas. "Novelty and Diversity in Recommender Systems." In: *Recommender Systems Handbook*. Ed. by F. Ricci, L. Rokach, and B. Shapira. Boston, MA: Springer US, 2015, pp. 881–918. ISBN: 978-1-4899-7637-6. DOI: 10.1007/978-1-4899-7637-6_26.

[20] L. Chen and P. Pu. "Critiquing-based recommenders: survey and emerging trends." In: *User Modeling and User-Adapted Interaction* 22.1 (Apr. 2012), pp. 125–150. ISSN: 1573-1391. DOI: 10.1007/s11257-011-9108-6.

[21] D. Cosley, S. K. Lam, I. Albert, J. A. Konstan, and J. Riedl. "Is Seeing Believing?: How Recommender System Interfaces Affect Users' Opinions." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '03. Ft. Lauderdale, Florida, USA: ACM, 2003, pp. 585–592. ISBN: 1-58113-630-7. DOI: 10.1145/642611.642713.

[22] L. W. Dietz, D. Herzog, and W. Wörndl. "Deriving Tourist Mobility Patterns from Check-in Data." In: *Proceedings of the WSDM 2018 Workshop on Learning from User Interactions*. 2018.

[23] G. Fischer. "User Modeling in Human–Computer Interaction." In: *User Modeling and User-Adapted Interaction* 11.1 (Mar. 2001), pp. 65–86. ISSN: 1573-1391. DOI: 10.1023/A:1011145532042.

[24] D. Gavalas, C. Konstantopoulos, K. Mastakas, and G. Pantziou. "Mobile recommender systems in tourism." In: *Journal of Network and Computer Applications* 39 (2014), pp. 319–333. ISSN: 1084-8045. DOI: https://doi.org/10.1016/j.jnca.2013.04.006.

[25] H. Gibson and A. Yiannakis. "Tourist roles: Needs and the Lifecourse." In: *Annals of Tourism Research* 29.2 (2002), pp. 358–383. ISSN: 0160-7383. DOI: https://doi.org/10.1016/S0160-7383(01)00037-8.

[26] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. "Using Collaborative Filtering to Weave an Information Tapestry." In: *Commun. ACM* 35.12 (Dec. 1992), pp. 61–70. ISSN: 0001-0782. DOI: 10.1145/138859.138867.

[27] L. R. Goldberg. "The structure of phenotypic personality traits." In: *American psychologist* 48.1 (1993), p. 26.

[28] M. Grčar, D. Mladenič, B. Fortuna, and M. Grobelnik. “Data Sparsity Issues in the Collaborative Filtering Framework.” In: *Advances in Web Mining and Web Usage Analysis*. Ed. by O. Nasraoui, O. Zaïane, M. Spiliopoulou, B. Mobasher, B. Masand, and P. S. Yu. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 58–76. ISBN: 978-3-540-46348-1.

[29] C. Grün, J. Neidhardt, and H. Werthner. “Ontology-Based Matchmaking to Provide Personalized Recommendations for Tourists.” In: *Information and Communication Technologies in Tourism 2017*. Ed. by R. Schegg and B. Stangl. Cham: Springer International Publishing, 2017, pp. 3–16. ISBN: 978-3-319-51168-9.

[30] D. Herzog and W. Wörndl. “A Travel Recommender System for Combining Multiple Travel Regions to a Composite Trip.” In: *CBRecSys@RecSys*. 2014.

[31] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich. *Recommender systems: an introduction*. Cambridge University Press, 2010.

[32] R. Kohavi, A. Deng, B. Frasca, T. Walker, Y. Xu, and N. Pohlman. “Online Controlled Experiments at Large Scale.” In: Aug. 2013. DOI: 10.1145/2487575.2488217.

[33] Y. Kurata. “CT-Planner2: More Flexible and Interactive Assistance for Day Tour Planning.” In: *Information and Communication Technologies in Tourism 2011*. Ed. by R. Law, M. Fuchs, and F. Ricci. Vienna: Springer Vienna, 2011, pp. 25–37. ISBN: 978-3-7091-0503-0.

[34] Y. Kurata and T. Hara. “CT-Planner4: Toward a More User-Friendly Interactive Day-Tour Planner.” In: *Information and Communication Technologies in Tourism 2014*. Ed. by Z. Xiang and I. Tussyadiah. Cham: Springer International Publishing, 2013, pp. 73–86. ISBN: 978-3-319-03973-2.

[35] C. Lamsfus, A. Alzua-Sorzabal, D. Martin, Z. Salvador, and A. Usandizaga. “Human-centric Ontology-based Context Modelling in Tourism.” In: *KEOD*. 2009, pp. 424–434.

[36] C. Laß, D. Herzog, and W. Wörndl. “Context-Aware Tourist Trip Recommendations.” In: *RecTour 2017* (2017).

[37] C. Laß, W. Wörndl, and D. Herzog. “A Multi-Tier Web Service and Mobile Client for City Trip Recommendations.” In: *Proceedings of the 8th EAI International Conference on Mobile Computing, Applications and Services*. MobiCASE’16. Cambridge, Great Britain: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2016, pp. 153–154. ISBN: 978-1-63190-137-9. DOI: 10.4108/eai.30-11-2016.2267194.

[38] P. Lops, M. de Gemmis, and G. Semeraro. "Content-based Recommender Systems: State of the Art and Trends." In: *Recommender Systems Handbook*. Ed. by F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor. Boston, MA: Springer US, 2011, pp. 73–105. ISBN: 978-0-387-85820-3. DOI: 10.1007/978-0-387-85820-3_3.

[39] S. Mangiafico. "Summary and analysis of extension program evaluation in R." In: *New Brunswick, NJ: Rutgers* 442 (2016).

[40] N. Manouselis and C. Costopoulou. "Analysis and Classification of Multi-Criteria Recommender Systems." In: *World Wide Web* 10.4 (Dec. 2007), pp. 415–441. ISSN: 1573-1413. DOI: 10.1007/s11280-007-0019-8.

[41] L. Mc Ginty and B. Smyth. "Deep dialogue vs casual conversation in recommender systems." In: (2002).

[42] K. McCarthy, M. Salamó, L. Coyle, L. McGinty, B. Smyth, and P. Nixon. "Group Recommender Systems: A Critiquing Based Approach." In: *Proceedings of the 11th International Conference on Intelligent User Interfaces*. IUI '06. Sydney, Australia: ACM, 2006, pp. 267–269. ISBN: 1-59593-287-9. DOI: 10.1145/1111449.1111506.

[43] L. McGinty and J. Reilly. "On the Evolution of Critiquing Recommenders." In: *Recommender Systems Handbook*. Ed. by F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor. Boston, MA: Springer US, 2011, pp. 419–453. ISBN: 978-0-387-85820-3. DOI: 10.1007/978-0-387-85820-3_13.

[44] I. Mínguez, D. Berrueta, and L. Polo. "CRUZAR." In: *Cases on Semantic Interoperability for Information Systems Integration*. IGI Global, 2010, pp. 255–271. DOI: 10.4018/978-1-60566-894-9.ch012.

[45] M. Montaner, B. López, and J. L. de la Rosa. "A Taxonomy of Recommender Agents on the Internet." In: *Artificial Intelligence Review* 19.4 (June 2003), pp. 285–330. ISSN: 1573-7462. DOI: 10.1023/A:1022850703159.

[46] A. Moreno, A. Valls, D. Isern, L. Marin, and J. Borràs. "SigTur/E-Destination: Ontology-based personalized recommendation of Tourism and Leisure Activities." In: *Engineering Applications of Artificial Intelligence* 26.1 (2013), pp. 633–651. ISSN: 0952-1976. DOI: https://doi.org/10.1016/j.engappai.2012.02.014.

[47] J. Neidhardt, L. Seyfang, R. Schuster, and H. Werthner. "A picture-based approach to recommender systems." In: *Information Technology & Tourism* 15.1 (Mar. 2015), pp. 49–69. ISSN: 1943-4294. DOI: 10.1007/s40558-014-0017-5.

[48] J. Nielsen. "10 usability heuristics for user interface design." In: *Nielsen Norman Group* 1.1 (1995).

[49] J. W. Payne, J. R. Bettman, and E. J. Johnson. *The adaptive decision maker*. Cambridge University Press, 1993.

[50] 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. Chicago, Illinois, USA: ACM, 2011, pp. 157–164. ISBN: 978-1-4503-0683-6. DOI: `10.1145/2043932.2043962`.

[51] F. Ricci and Q. N. Nguyen. "Acquiring and Revising Preferences in a Critique-Based Mobile Recommender System." In: *IEEE Intelligent Systems* 22.3 (May 2007), pp. 22–29. ISSN: 1541-1672. DOI: `10.1109/MIS.2007.43`.

[52] F. Ricci, L. Rokach, and B. Shapira. "Recommender Systems: Introduction and Challenges." In: *Recommender Systems Handbook*. Ed. by F. Ricci, L. Rokach, and B. Shapira. Boston, MA: Springer US, 2015, pp. 1–34. ISBN: 978-1-4899-7637-6. DOI: `10.1007/978-1-4899-7637-6_1`.

[53] P. J. Rousseeuw. "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis." In: *Journal of Computational and Applied Mathematics* 20 (1987), pp. 53–65. ISSN: 0377-0427. DOI: `https://doi.org/10.1016/0377-0427(87)90125-7`.

[54] T. Ruotsalo, K. Haav, A. Stoyanov, S. Roche, E. Fani, R. Deliai, E. Mäkelä, T. Kauppinen, and E. Hyvönen. "SMARTMUSEUM: A mobile recommender system for the Web of Data." In: *Web Semantics: Science, Services and Agents on the World Wide Web* 20 (2013), pp. 50–67. ISSN: 1570-8268. DOI: `https://doi.org/10.1016/j.websem.2013.03.001`.

[55] G. Salton, A. Wong, and C. S. Yang. "A Vector Space Model for Automatic Indexing." In: *Commun. ACM* 18.11 (Nov. 1975), pp. 613–620. ISSN: 0001-0782. DOI: `10.1145/361219.361220`.

[56] A. Savir, R. Brafman, and G. Shani. "Recommending Improved Configurations for Complex Objects with an Application in Travel Planning." In: *Proceedings of the 7th ACM Conference on Recommender Systems*. RecSys '13. Hong Kong, China: ACM, 2013, pp. 391–394. ISBN: 978-1-4503-2409-0. DOI: `10.1145/2507157.2507196`.

[57] I. Seidel, M. Gärtner, M. Pöttler, H. Berger, and M. Dittenbach. "Itchy feet: a 3D e-tourism environment." In: *Tourism Informatics: Visual Travel Recommender Systems, Social Communities, and User Interface Design*. IGI Global, 2010, pp. 209–242.

[58] A. Sieg, B. Mobasher, and R. Burke. "Web Search Personalization with Ontological User Profiles." In: *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*. CIKM '07. Lisbon, Portugal: ACM, 2007, pp. 525–534. ISBN: 978-1-59593-803-9. DOI: `10.1145/1321440.1321515`.

[59] P. Vansteenwegen, W. Souffriau, and D. Van Oudheusden. "The orienteering problem: A survey." In: *European Journal of Operational Research* 209.1 (2011), pp. 1–10.

[60] R. E. Walpole, R. H. Myers, S. L. Myers, and K. Ye. *Probability and statistics for engineers and scientists*. Vol. 5. Macmillan New York, 1993.

[61] W. Wörndl. "A Web-based Application for Recommending Travel Regions." In: *Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization*. UMAP '17. Bratislava, Slovakia: ACM, 2017, pp. 105–106. ISBN: 978-1-4503-5067-9. DOI: 10.1145/3099023.3099031.

[62] WTTC. *Travel & Tourism Economic Impact 2018*. https://www.wttc.org/-/media/files/reports/economic-impact-research/regions-2018/world2018.pdf. Accessed: 2019-01-13.

[63] A. Yiannakis and H. Gibson. "Roles tourists play." In: *Annals of Tourism Research* 19.2 (1992), pp. 287–303. ISSN: 0160-7383. DOI: https://doi.org/10.1016/0160-7383(92)90082-Z.