

Article

Enhancing Recommender System with Collaborative Filtering and User Experiences Filtering

Silvana Vanesa Aciar ^{1,*}, Ramon Fabregat ² , Teodor Jové ² and Gabriela Aciar ³

¹ Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Universidad Nacional de San Juan, San Juan 5400, Argentina

² Broadband Communications and Distributed Systems, Institut de Informàtica i Aplicacions, Universitat de Girona, 17004 Girona, Spain; ramon.fabregat@udg.edu (R.F.); teodor.jove@udg.edu (T.J.)

³ Instituto de Informática, Universidad Nacional de San Juan, San Juan 5400, Argentina; gaby_aciar@yahoo.com.ar

* Correspondence: silvana.aciar@conicet.gov.ar

Abstract: Recommender systems have become an essential part in many applications and websites to address the information overload problem. For example, people read opinions about recommended products before buying them. This action is time-consuming due to the number of opinions available. It is necessary to provide recommender systems with methods that add information about the experiences of other users, along with the presentation of the recommended products. These methods should help users by filtering reviews and presenting the necessary answers to their questions about recommended products. The contribution of this work is the description of a recommender system that recommends products using a collaborative filtering method, and which adds only relevant feedback from other users about recommended products. A prototype of a hotel recommender system was implemented and validated with real users.

Keywords: recommender system; collaborative filtering; opinion mining; user experience



Citation: Aciar, S.V.; Fabregat, R.; Jové, T.; Aciar, G. Enhancing Recommender System with Collaborative Filtering and User Experiences Filtering. *Appl. Sci.* **2021**, *11*, 11890. <https://doi.org/10.3390/app112411890>

Academic Editors:
Ángel González-Prieto and
Fernando Ortega

Received: 31 October 2021
Accepted: 8 December 2021
Published: 14 December 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Recommender systems have become an essential part in many e-commerce sites, entertainment sites, tourism applications and others to address the problem of information overload [1,2]. Currently, these systems automatically present products, services, content or information that may be of interest to users [1–4]. Most recommender systems rely on the historical ratings of users to obtain a list of preferences. Despite being widely used, these systems suffer from the problem of scarcity of data when users do not explicitly provide feedback on the recommendations received [2,3].

In recent years, new trends and proposals have emerged to improve the effectiveness of recommendation algorithms by improving their methods or exploring new sources of information to obtain user preferences [1,3,4]. Currently, machine learning methods are used to process the huge preference matrices. These methods operate as black boxes that generally receive numbers as input information and output other numbers that represent predictions of preferences on the items to be recommended [3,5].

Although in current recommender systems such as those used by Netflix, Amazon, Booking.com and others, users receive only relevant products as recommendations, these users review the opinions of others before accepting the recommended product.

Generally, opinions regarding a given product are numerous and a user can spend hours looking for relevant information. The objective of this work is to present opinions relevant to a particular recommended product. Specifically, once the product has been recommended, the most relevant excerpts are retrieved from the opinions of other users to help the user in the process of deciding whether to accept the recommended product. For example, Netflix recommends movies, but users like to read the opinions of others

before watching recommended movies. These sentences contain information about the experiences of other users who have seen the movie, i.e., feelings, context, target users and general evaluations of the film.

The contribution of this work is a recommender system that recommends products using any collaborative filtering method, and presents only relevant opinions from other users about the recommended product. A prototype of a hotel recommender system was implemented.

The rest of the article is structured as follows. A survey to obtain information from user opinions is discussed in Section 2. Related works are presented in Section 3. Section 4 presents a previous work used in our present proposal. Section 5 shows the recommendation method proposal in which information about relevant user experiences of the recommended items is added. A case study to validate the proposal is detailed in Section 6. Finally, in Section 7, a conclusion and future work are presented.

2. Survey

A survey was conducted to determine whether users read opinions, even if a product has been recommended to them. The survey was conducted with 350 people from different countries, such as Spain, Colombia, Argentina, Brazil, Portugal, Italy, France, Belgium, Chile, Uruguay and England. Ten questions were asked, as shown in Table 1.

Table 1. Questions and results of the survey of users of recommender systems.

Question	Response Value	Result
1. Do you use platforms that recommend products, hotels, movies or other services or information?	Yes/No	99%
2. Name the platforms you use	Free text	Netflix, Amazon, Booking, AirNbn, Facebook, Instagram, Mercado Libre, Pinterest, Spotify,
3. Do you read the opinions of other users when you receive movie recommendations from these platforms?	Yes/No	91%
4. Do you read the opinions of other users when you receive product recommendations from these platforms?	Yes/No	98%
5. Do you read the opinions of other users when you receive hotel recommendations from these platforms?	Yes/No	99%
6. Do you read the opinions of other users when you receive recommendations of places to travel from these platforms?	Yes/No	99%
7. Do you read the opinions of other users when you receive restaurant recommendations from these platforms?	Yes/No	99%
8. Do you read the opinions of other users when you receive recommendations for leisure or entertainment activities from these platforms?	Yes/No	99%
9. How long does it take you to read opinions of recommended products? Describe the average time.	Time in hours, minutes and seconds	1.50 h (Total average time from all surveyed) - To learn about the quality of the product - To determine if the article is really useful for your needs - To determine whether the seller is honest about the functionality of the product
10. Why do you read the opinions of recommended products?	Free text	- To learn about the experiences that other buyers had - To learn about the negative characteristics of the product - To learn about the context or conditions of use

The results presented in the table show that 98% of respondents read reviews of recommended products before buying. Specifically, 91% read movie reviews on the most popular streaming platforms and 99% rely on the opinions of other users before booking a hotel, restaurant or leisure activity. In the survey, users were asked about the amount of time they spend searching for and reading reviews of recommended products. The average time is 1.50 h. Participants in the survey noted that they spend around 15 min reading movie reviews to 4 h reading reviews of hotels and leisure activities. Analyzing the reasons for reading the opinions of recommended products, they were grouped and summarized as follows: to learn about the quality of the product; to determine if the article is really useful for your needs; to determine whether the seller is honest about the functionality of the product; to learn about the experiences that other buyers had; to learn about the negative characteristics of the product; and to learn about the context or conditions of use.

To summarize the results of the survey, we highlight that:

1. People who use platforms where products are recommended read opinions regarding recommended products before buying them.
2. For each recommended product, they spend up to 1.5 h reading opinions.
3. They read user feedback to learn about the conditions of use of the product, the quality of the product, the reliability of the seller, to obtain a general overview of the product, to learn about the negative characteristics of the product, among others.
4. It is necessary to provide recommender systems with methods that present information about the experiences of other users along with the presentation of the product itself. These methods should help users by filtering reviews and presenting only the necessary information.

3. Related Works

Currently, information generated by users is being explored and linked to recommendations either by expanding the filtering method or as an alternative method to solve the problem of data scarcity. For example, in [6], long short-term memory (LSTM) is used to extract relevant aspects of user reviews. The authors of that study define three types of sentences, i.e., those that contain context words, those that contain words associated with the characteristics of the item, and those that contain polarization words (good, bad, etc.). Mutual information mechanisms are used to integrate the different classes into a single, final opinion. Gradient reinforcement learning is used to minimize errors. A framework based on deep learning is proposed in [7] to generate explanations of recommendations from reviews about target items. The proposal was validated with data from Amazon.com. The incorporation of opinions in the recommendation system, as a means of providing additional feedback, is proposed in [8]. A comparison of various collaborative filtering methods demonstrated that the value decomposition (SVD), non-negative matrix factorization (NMF), and SVD ++, used with a combination of explicit user ratings as well as ratings derived from opinions, improves the effectiveness of recommender systems. In [9], a methodology to obtain explanations of recommended items is proposed. The most relevant aspects discussed in the opinions are obtained. Then, positive sentences associated with those aspects are retrieved. Finally, a template is filled out with positive information, which is then presented to users as a justification of why they should obtain the item in question. Natural language processing techniques are used. In [10], the authors analyzed opinions to generate explanations of recommendations using a multilevel dynamic programming algorithm called Deep Explicit Attentive Multi-View Learning (DEAML), in addition to predicting the ratings from opinions. A recurrent neural network to generate explanations of recommendations from textual and visual information is used in [11]. The authors of that study presented which ever aspect of an image they focused on to make recommendations. The authors in [12] proposed a multiple view recurrent neural network model for sequential recommendations in which a matrix representing the user's ratings is multidimensional, containing information from different sources, including user opinions.

In [13], a thoughtful, aspect-based recommendation model that effectively captures the interactions between aspects extracted from reviews is presented.

In recent years, popular services such as Netflix have applied matrix factorization methods to make recommendations. Matrix factorization has been shown to be effective for processing large amounts of information, but ascarcity of data makes it deficient [14,15]. Explicit user feedback to predict ratings is increasingly difficult to obtain. Works related to obtaining implicit feedback from users have emerged as a solution to this problem [16]. Generally, the text is transformed into a numerical value to be treated as an explicit rating in collaborative filtering matrices. This action generates uncertainty and error in the transformation of a large amount of text to a single numerical value [17]. A matrix factorization method that uses implicit user information was proposed in [18]. The implicit information comes from a list of products that have been viewed by a user, who went on to view a target product. In [19], an implicit rating value is attributed to the usefulness of a given opinion; the authors of that study propose a neural attentional regression model to predict more precise ratings. In [20], rating sare generated by combining explicit and implicit feedback, and a probabilistic matrix factoring model is used to predict that rating. Explicit feedback is when a user rates an item, and when other users indicate that that the opinion was useful. The search history is considered implicit feedback. In [21], two matrices were constructed. One contains explicit negative feedback and the other the viewing history as implicit feedback. A neural model was used to predict ratings.

Based on our analysis of the related works, it was concluded that opinions may be used to obtain additional ratings about items, or to explain the reasons for recommending certain products. However, no research was found presenting explicit answers to questions such as the following: Is the product useful? Does it have a defect? In what situations is it usable? Is it for everyone? These questions are what users ask themselves when reading product opinions before purchasing.

4. Previous Works

In our previous work [22], a method to retrieve relevant sentences from opinions was defined. A set of categories to group relevant sentences was established, as well as a set of rules to classify sentences into these categories. The defined categories were:

Definition 1. *Evaluation: Information to evaluate the item or parts of it.*

Definition 2. *Sensation: Information that represents the effect, sensation or emotion caused by the item to the user.*

Definition 3. *Recommendation: Information where the user suggests or recommends the item and considers it advantageous, beneficial, adequate or timely for useonsome occasion or situation.*

Definition 4. *Target users: Information about people, animals, entities, etc. that are related to the user or item and those whoparticipated in the experience.*

This kind of information answers the questions posedby users before buying an item, as can be concluded from the survey we carried out. The set of rule defined in [22] is the following:

- R1. w_i = "for" or " for" + nou. person or "to" + verb. motion -> Recommendation
- R2. Syntactic_cat (w_i) = Ad. all -> Evaluation
- R3. Syntactic_cat (w_i) = verb. emotion -> Sensation
- R4. Syntactic_cat (w_i) = verb. perception -> Sensation
- R5. Syntactic_cat (w_i) = verb. change. -> Sensation
- R6. Syntactic_cat (w_i) = nou. artifact -> Recommendation
- R7. Syntactic_cat (w_i) = noun. location -> Recommendation
- R8. Syntactic_cat (w_i) = verb. weather -> Recommendation

- R9. Syntactic_cat (wi) = verb. motion -> Recommendation
- R10. Syntactic_cat (wi) = verb.communication -> Recommendation
- R11. Syntactic_cat (wi) = noun.communication -> Recommendation
- R12. Syntactic_cat (wi) = noun. act -> Recommendation
- R13. Syntactic_cat (wi) = verb. consumption -> Recommendation
- R14. Syntactic_cat (wi) = noun. time -> Recommendation
- R15. Syntactic_cat (wi) = verb. social, -> Recommendation
- R16. Syntactic_cat (wi) = noun. cognition -> Recommendation
- R17. Syntactic_cat (wi) = noun. person -> Target_User
- R18. Syntactic_cat (wi) = noun. group -> Target_User
- R19. Syntactic_cat (wi) = noun. animal -> Target_User

In the present work, the rules are only used to retrieve relevant sentences concerning recommended products.

5. Proposal: A Recommender System That Recommends Products and a List of Relevant Sentences from User Reviews

A process that, upon receiving a query from a user, recommends an item and retrieves the most relevant user experiences from user reviews is proposed. Reviews can contain many sentences that are not relevant. Figure 1 presents the process proposed to achieve this purpose.

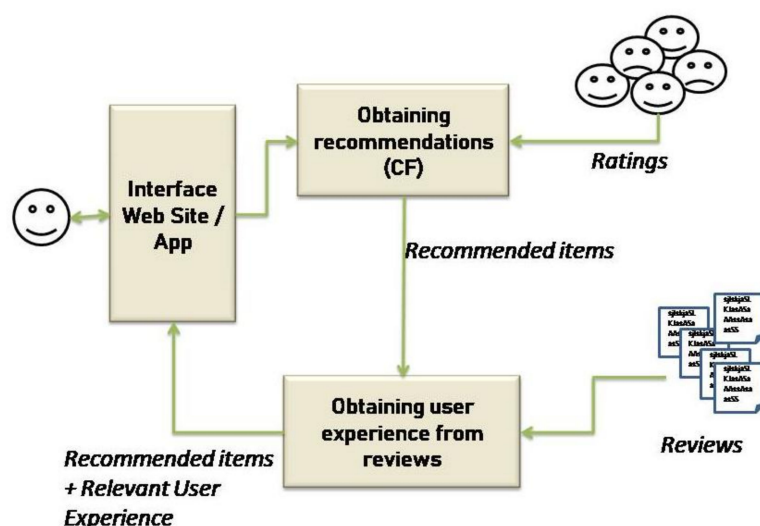


Figure 1. Recommending items and sentences containing relevant user experiences.

The user enters a query into the system. Then, recommended items are obtained through a CF method. The ratings of users with similar preferences are used to predict the ratings of other items to recommend. The items to be recommended are ordered according to the values of their predicted ratings. Then, the first 10 items are recommended to the user. For each 10 items, the opinions of other users are retrieved. A process to obtain the experiences that other users had with that item is applied to the retrieved opinions. Only the most relevant experiences are presented to the user. Details of each step in this process are presented in the following subsections.

5.1. Obtaining Recommendations

As the objective of this work is to provide recommender systems with methods that display information about the experiences of other users along with the presentation of recommended products, the method should work with whichever collaborative filtering (CF) technique is currently being used on the recommender platform. That is why in our proposal, we begin by analyzing the techniques which are most widely used in CF to make

recommendations. Then, these techniques, contained in a data set, are implemented and the most efficient one is selected for use as a real recommender system prototype.

CF is the most widely used method in RS. It is widely used in the most successful e-commerce sites and applications, such as Netflix (<https://www.netflix.com>), Amazon (<https://www.amazon.com>), Facebook (<https://www.facebook.com>), Booking (<https://www.booking.com>), Airbnb (<https://www.airbnb.com>).

However, CF techniques suffer from certain problems [1,23] such as data dispersion, cold start and scalability issues, which affect the precision of the recommendations [1,5]. Methods that model and process information from social networks and user reviews, and then incorporate them as parameters into the recommendation metrics, were proposed in [2,23]. Regarding scalability problems, solutions based on machine learning have been proposed, such as clustering (KNN), singular value decomposition (SVD), support vector machines (SVM) and neural networks (NN), among others [1,2,5,23].

In this proposal, three of the most popular CF methods (SVM, KNN and SVD) were analyzed to select the one that best predicted recommendations. The recommendation methods were implemented with a set of hotel data. Predictions were subsequently evaluated and the obtained results were compared. The application of the three methods is detailed below.

The implementation used a data set available at <https://data.world/datafiniti/hotel-reviews>. This data set has rating information made by 1434 users regarding 1000 hotels in the United States. It also has the opinions of those users, which is useful for the purpose of this research.

The methods were implemented in Python language. The data set was divided into two groups as follows: 70% of the randomly selected data were used to train the algorithms, and the remaining 30% were used for validation. A series of experiments were carried out to see how the precision of the recommendations of the different methods varied.

5.1.1. K Nearest Neighbors (KNN)

The objective of this method is to group users or items with similar preferences and interests. Based on similarities, a rating r' is predicted for an item which has not yet been valued [23,24]. That is, given users $u \in U$, then k users v closest to u (called the neighborhood of u $N(u)$) are obtained, and rating r'_{ui} is obtained from the ratings from users $v \in N(u)$ (Equation (1)).

$$r'_{ui} = w_u + \frac{\sum_{v \in N} \text{sim}(u, v)(r_{vi} - w_v)}{\sum_{v \in N} \text{sim}(u, v)} \quad (1)$$

Rating r' is the weighted sum of the ratings of the k most similar users, where w is the average score of each user u .

In this work, Pearson's correlation (Equation (2)) was used in the KNN method to obtain the 50 closest neighbors.

$$\text{sim}(u, v) = \frac{\sum_{i \in I} (r_{ui} - w_u)(r_{vi} - w_v)}{\sqrt{\sum_{i \in I} (r_{ui} - w_u)^2 \sum_{i \in I} (r_{vi} - w_v)^2}} \quad (2)$$

5.1.2. Support Vector Machine (SVM)

ASVM is basically a binary classifier that tries to find a hyper plane in a vector space that better separates two classes in a data set. Such a hyper plane can be represented as a set of points x that satisfy $\langle w, \phi(x) \rangle - b = 0$, where $\langle \rangle$ is the inner product, w represents a vector normal to the hyper plane, b represents the distance of the plane from the origin and ϕ is a kernel function. For a new data point, SVM decides its class based on which side of the hyper plane the new data point is on [23,25]. In this work, as it is a multiclass classification problem, a combination of SVM linear binary classifiers was used for the classification of five classes, i.e., the five possible rating values.

5.1.3. Singular Value Decomposition (SVD)

SVD is a matrix factorization method where the dimensions of the component space of a matrix are reduced. The rating matrix is decomposed into two matrices; the product of these matrices is the original matrix. Each user is represented by a vector p , the item by a vector q , and the rating matrix R is the scalar product of both vectors p and q (Equation (3)) [26,27].

$$r'_{ui} = \mu + b_u + b_i + p_u q_i \quad (3)$$

where μ is the average rating of the items, b_i is the average rating of item i minus μ , and b_u is the average rating given by the user u minus μ . The SDG gradient descent method was used to minimize the equation where the step size γ was set at 0.005.

5.1.4. Validation of KNN, SVM and SVD Implemented in the CF

The mean square error (MSE) method is used to obtain the degree of error when predicting unknown ratings (Equation (4)). The precision, recall and FScore measures (Equations (5), (6) and (7), respectively) were used to evaluate the efficacy of the methods in obtaining relevant recommendations [23,28,29].

$$MSE = \frac{\sum_{j=1}^n (r'_{ui} - r_{ui})^2}{n} \quad (4)$$

where r'_{ui} is the predicted rating, r_{ui} is the real rating, and n is the number of predictions.

The precision (Equation (5)) is the number of ratings predicted as being relevant (t) from among all recommended ratings (n).

$$Precision = \frac{t}{n} \quad (5)$$

Recall (Equation (6)) is the number of ratings predicted as being relevant (t) divided by the total ratings considered to be relevant for a user (m).

$$Recall = \frac{t}{m} \quad (6)$$

The FScore is a single value combining the Precision and Recall values. If the FScore is 1, Precision and Recall have the best values.

$$FScore = \frac{2(Precision * Recall)}{Precision + Recall} \quad (7)$$

These measures were used to compare the recommendation methods implemented in the hotel ratings data set. Predicted ratings with a value greater than 4 were considered relevant for calculating Precision and recall. Table 2 presents the values of the evaluation measures of the three methods.

Table 2. Evaluation results of CF methods.

Methods	MSE	Precision	Recall	FScore
SVM	0.657	0.741	0.763	0.751
KNN	0.578	0.787	0.806	0.796
SVD	0.456	0.836	0.851	0.843

Table 2 presents a performance analysis of the three analyzed methods. The results obtained with the SVD method were better than those from the other methods. SVD achieved the lowest MSE value, as well as a relatively good FScore, Recall and Precision, which shows that the matrix factorization in our data set benefited the CF results.

5.2. Extracting User Experiences

Once the system has selected products to recommend, it must retrieve the relevant sentences to address questions such as the following: How good is the product? How did you feel when you used it? Do you recommend the product? Under what conditions is it used? Who else used it? The answers to these questions represent the kind of information that a user looks for before deciding to purchase an item, even if it was automatically recommended.

The set of rules defined in our previous work was used to obtain relevant sentences regarding products to be recommended. The rules were validated in our data set (<https://data.world/datafiniti/hotel-reviews>) in order to determine if they were an efficient method by which to obtain relevant sentences. Half of the 10,000 reviews were randomly selected to create a data set which was manually labeled with user experience categories. Fifty-five people manually tagged each sentence in one of the user experience categories: Evaluation, Sensation, Recommendation and Target Users. A group of teachers, university researchers and graduate students of universities in Argentina, Brazil, Chile, France, Portugal and Spain, aged between 25 and 57, labeled the opinions. The total number of classified sentences was 16,755. Each sentence was tagged by three people. The sentences that failed to achieve consensus were reclassified by another three people, and a voting system was used to categorize them.

In this data set, rules were implemented to classify each sentence into one category of user experience. Support and confidence measures were used for each of the rules [30]. Since the rules had the form of $X \Rightarrow Y$, support was defined as the number of times the rule in the analyzed sentences had been met. Confidence was defined as the number of times that X and Y appeared together in sentences containing X . Table 3 shows the support and confidence values for each of the defined rules.

Table 3. Support values and confidence of the rules applied to the hotel set data set.

Rules	Support	Confidence
R1	0.77	0.95
R2	0.85	0.96
R3	0.72	0.74
R4	0.97	0.98
R5	0.8	0.86
R6	0.73	0.97
R7	0.82	0.94
R8	0.75	0.88
R9	0.84	0.96
R10	0.85	0.74
R11	0.72	0.98
R12	0.73	0.97
R13	0.66	0.94
R14	0.86	0.96
R15	0.73	0.74
R16	0.68	0.98
R17	0.88	0.98
R18	0.75	0.85
R19	0.56	0.75

In addition, the VSM method was used in our data set to classify the sentences in the user experience category, and to compare its efficiency regarding classifications based on the rules. A combination of SVM linear binary qualifiers was used for the classification of opinions in multiple classes.

The four classes of sentences were defined as Evaluations (E), Sensations (S), Recommendations (R) and Target Users (U). The data set was divided into two parts: 80% for training and 20% for testing. The inputs to the classifiers were numerical vectors. The labeled sentences were transformed into vectors using the word bag strategy and accord-

ing to their frequency. Three binary classifiers were defined using the following strategy (Directed Acyclic Graph SVM): E and U, E and R, E and S. In this way, the output was the label belonging to one of the trained classes.

The MSE measure (Equation (4)) was used to measure the error between the predicted classification and the manual classification carried out in the implementation of both methods. Table 4 shows the MSE values obtained from the classification of sentences in both methods.

Table 4. MSE values of the lexical rules and SVM.

Methods	MSE
SVM	0.589
Lexical Rules	0.364

The method based on lexical rules had the lowest error value. It classified better than SVM. This is because with the rules-based method, a deeper analysis was performed, i.e., the internal structures of the sentences were analyzed. We found cases where a sentence belonged to more than one category, and as such, could not be classified.

5.3. Final Recommendations

The final step was to make recommendations. The 10 items with the highest predicted rating values were recommended to the user. For each of the 10 items, user opinions were recovered and sentences from the user experience category were obtained. Six sentences were displayed based on the categories of user experience. The ranking system took into account the “*level of the reviewer*”. The level of the reviewer was given by the number of opinions he/she had written (Equation (8)).

$$\text{Level of the reviewer} = \frac{COO}{COT} \quad (8)$$

where COO is the number of opinions that this user has made and COT is the total number of opinions made by all users.

In summary, the user receives a list of items; for each item, a set of the most relevant sentences in each category is presented, rather than a list of opinions containing information that may not be relevant.

6. Case of Study

A prototype hotel recommender app was developed for the purpose of validating the presented proposal. We must emphasize that this app was developed and used only for research and evaluation purposes. This app allowed users to obtain hotel recommendations upon submitting a query. For example, if a user enters “Hotel in London”, the app returns 10 London hotels and relevant excerpts from reviews, ranked by category of user experience. Figure 2 provides information about one of the recommended hotels.

Hotels are recommended by applying the FC method described in Section 5.1. Opinions classified by user experience are obtained using the rules and filtering methods presented in Sections 4 and 5.3, respectively. Users must rate at least 10 hotels which they have visited the first time they enter the system. The system allows users to search for 10 or more previously visited hotels and give them a rating. The history of user ratings is built with this information, avoiding the cold start problem. Also, the system builds the rating matrix of the hotels in the system by taking the explicit ratings given by users on the TripAdvisor website. More specifically, ratings from 1 to 5 given by users are obtained for the construction of the rating matrix of the hotels in the system. The matrix of the hotels in the location chosen by the user and the historical ratings of that user are used to calculate the FC.

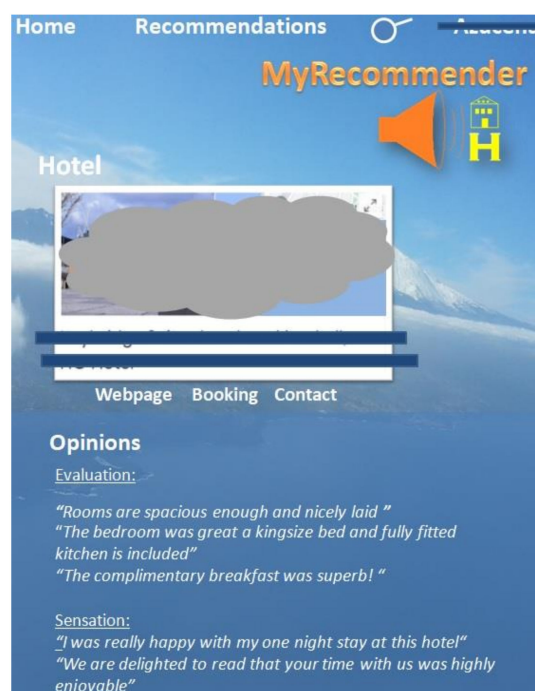


Figure 2. Recommended hotel and relevant sentences from user feedback.

Opinions regarding recommended hotels were obtained from various websites, such as Tripadvisor.com and Booking.com. We did this only for research purposes and the development of a prototype in a controlled environment.

Seventy people from countries such as Argentina, Brazil, Australia, Colombia, Belgium, Spain, France and Portugal participated in the validation. The app was used by 41 women and 29 men, aged from 26 to 58. Participants were required to enter a query about a location they intended to visit. Then, the system presented hotels in that location. A total of 320 queries were made in the system. After receiving the recommendations, users had to rate their level of satisfaction on a scale from 1 (not at all satisfied) to 5 (very satisfied).

Validation

Two aspects were validated: user satisfaction with the system and the precision of the recommendations. A precision measure based on user satisfaction was used to evaluate the received recommendations (Equation (5)), where t is the number of recommendations with a user satisfaction value > 4 and n is the total number of recommendations made.

Validation of the overall performance of the system was carried out using the Technological Acceptance Model (TAM) method [31–33]. The TAM model is used to validate the adoption of technology by users. This model serves to predict the use of a technology based on two main features: (1) perceived utility (PU) and (2) perceived ease of use (PEU). PU refers to the degree of performance improvement in a task when using the system. PEU is the degree of user effort required to perform tasks.

In this study, TAM was used to measure the degree of acceptance that users had with regard to the recommendations and the way in which they were presented. A questionnaire was designed to evaluate the performance of the proposed system according to the TAM method. The questionnaire and the results, summarized for each question put to the 70 participants, are presented in Table 5.

Table 5. Questionnaire and results of the validation using the TAM method.

Question	Yes	Not	Undecided
1. Is the system user-friendly?	70		
2. Are the organization and navigation structure adequate?	70		
3. Was information overload avoided?	69		1
4. Were the query criteria sufficient?	67	3	
5. Are the results displayed in a compressible way?	70		
6. Does the system provide assistance in case of not being able to offer results?	70		
7. Are you satisfied with the received recommendations?	66	4	
8. Were the received opinions useful?	70		
9. Do you want the system to automatically present relevant opinions?	70		
10. Was the recommendation of hotels with relevant opinions helpful to you?	70		
11. Do you consider the classification of opinions in these categories to be accurate and coherent?	70		
12. Was the information received by category of user experience useful?	69	1	
13. Would you use the APP in the future?	70		
14. Would you recommend the system to others?	70		

The Cronbach Alpha coefficient was used to measure the reliability of the TAM method [33–35]. Cronbach's alpha is an index of internal consistency that takes values between 0 and 1 and verifies whether the instrument that is being evaluated collects defective information, which would lead to wrong conclusions. Cronbach's alpha is calculated using Equation (9).

$$\alpha = \frac{K}{K-1} \left[1 - \frac{\sum i}{Vi} \right] \quad (9)$$

where α is the alpha of Cronbach, K is the number of items within the criterion to be evaluated, i is the variance of each item and Vi the total variance. Good reliability is indicated with a value >0.70 . The Cronbach coefficient for the survey was 0.9798. From this value, we can confirm that the questionnaire was reliable, that is, it effectively measured what it intended to measure. Next, Cronbach's alpha was also calculated for each of the criteria of the TAM method separately. Alpha = 0.9592 was obtained for PU, and Alpha = 0.9875 was obtained for PEU. These values allowed us to ensure that results of the questionnaire based on the TAM method were correct.

The results of the validation show that, in general, users were very satisfied with the proposed system, i.e., 100% of users rated the system interface, navigation and structure favorably. However, three users noted that the query criteria were not sufficient. The reasons described by them were that it was necessary to apply an advanced search mechanism with some other criteria. The current query only allowed users to enter text with the name of a city or country where they wanted to travel. Regarding the most important functionality of the system, i.e., providing recommendations with information associated with the experience of other users, this was rated as satisfactory by the majority of users. Four users were not satisfied with the received recommendations. They argued that the recommended hotels did not meet their preferences. When analyzing this problem in more detail, it was observed that they had interacted with the system once, and the system did not have information to obtain more precise preferences.

7. Conclusions

The objective of this work was to improve the interaction of users with recommendation systems. While the function of recommender systems is to solve the problem of

information overload in presenting relevant items to users, the user also consults feedback provided by other users before deciding to acquire the recommended item. The number of opinions available is considerable, so a user may take a long time to analyze such opinions, and this diminishes the effectiveness of the recommendations. In our proposal, the user experience is not presented as a list of opinions; rather, it is filtered, presenting only relevant sentences by grouping them into four categories: those expressing sensations, those evaluating the item, those expressing the target user, and those describing situations in which the item was used.

A study case with real users was implemented to recommend hotels and other users' experiences about those hotels. The results of the validation showed that users feel more satisfied when receiving filtered information regarding other people's experiences of using the recommended items.

Various aspects of our proposal will be improved in future work, such as: (1) analyzing how to represent user experience categories in a numerical vector; (2) integrating information from user experiences as another parameter in collaborative filtering models; (3) implementing and validating multilabel classification methods for sentences in various user experience categories (in some cases, a sentence may belong to two or three categories); (4) implementing other methods for classifying sentences in user experience categories such as fast Text [36], EL Mo [37] and Bert [38]; and (5) integrating a value of penalization based on negative opinions of the product into the recommender ratings.

Author Contributions: Conceptualization, S.V.A., R.F. and T.J.; funding acquisition, R.F.; investigation, S.V.A., R.F. and G.A.; methodology, S.V.A. and G.A.; supervision, R.F. and T.J.; writing—original draft, S.V.A., G.A. and R.F.; writing—review and editing, S.V.A., R.F. and T.J. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the DURSI consolidated research group Smart IT Engineering and Services (SITES) (ref. 2017 SGR-1551), Institute of Informatics and Applications at the University of Girona, and by and by the National Council for Scientific and Technical Research (CONICET), Argentina.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Fkih, F. Similarity measures for Collaborative Filtering-based Recommender Systems: Review and experimental comparison. *J. King Saud Univ.-Comput. Inf. Sci.* **2021**. [CrossRef]
2. Srifi, M.; Oussous, A.; AitLahcen, A.; Mouline, S. Recommender Systems Based on Collaborative Filtering Using Review Texts—A Survey. *Information* **2020**, *11*, 317. [CrossRef]
3. Zhang, S.; Yao, L.; Sun, A.; Tay, Y. Deep learning based recommender system: A survey and new perspectives. *ACM Comput. Surv. (CSUR)* **2019**, *52*, 1–38. [CrossRef]
4. Rose, C.; Cohen, W. Transnets: Learning to Transform for Recommendation. In Proceedings of the Eleventh ACM Conference on Recommender Systems, Como, Italy, 27–31 August 2017; pp. 288–296.
5. Haruna, K.; Ismail, M.A.; Suhendroyono, S.; Damiasih, D.; Pierewan, A.C.; Chiroma, H.; Herawan, T. Context-aware recommender system: A review of recent developmental process and future research direction. *Appl. Sci.* **2017**, *7*, 1211. [CrossRef]
6. Yang, M.; Qu, Q.; Shen, Y.; Liu, Q.; Zhao, W.; Zhu, J. Aspect and Sentiment Aware Abstractive Review Summarization. In Proceedings of the 27th International Conference on Computational Linguistics, Santa Fe, NM, USA, 20–26 August 2018; pp. 1110–1120.
7. Zarzour, H.; Jararweh, Y.; Hammad, M.M.; Al-Smadi, M. A Long Short-Term Memory Deep Learning Framework for Explainable Recommendation. In Proceedings of the 2020 11th International Conference on Information and Communication Systems (ICICS), online, 7–9 April 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 233–237.
8. Dang, C.N.; Moreno-García, M.N.; Prieta, F.D.L. An Approach to Integrating Sentiment Analysis into Recommender Systems. *Sensors* **2021**, *21*, 5666. [CrossRef] [PubMed]
9. Musto, C.; Lops, P.; de Gemmis, M.; Semeraro, G. Justifying Recommendations through Aspect-Based Sentiment Analysis of Users Reviews. In Proceedings of the 27th ACM Conference on User Modeling, Adaptation and Personalization, Larnaca, Cyprus, 9–12 June 2019; pp. 4–12.

10. Gao, J.; Wang, X.; Wang, Y.; Xie, X. Explainable Recommendation through Attentive Multi-View Learning. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; pp. 3622–3629.
11. Liu, P.; Zhang, L.; Gulla, J.A. Dynamic attention-based explainable recommendation with textual and visual fusion. *Inf. Process. Manag.* **2019**, *6*, 102099. [\[CrossRef\]](#)
12. Cui, Q.; Wu, S.; Liu, Q.; Zhong, W.; Wang, L. MV-RNN: A multi-view recurrent neural network for sequential recommendation. *IEEE Trans. Knowl. Data Eng.* **2018**, *32*, 317–331. [\[CrossRef\]](#)
13. Guan, X.; Cheng, Z.; He, X.; Zhang, Y.; Zhu, Z.; Peng, Q.; Chua, T.-S. Attentive aspect modeling for review-aware recommendation. *ACM Trans. Inf. Syst.* **2019**, *37*, 1–27. [\[CrossRef\]](#)
14. Abdi, M.; Okeyo, G.; Mwangi, R.W. Matrix factorization techniques for context-aware collaborative filtering recommender systems: A survey. *Comput. Inf. Sci.* **2018**, *11*, 1–10. [\[CrossRef\]](#)
15. Park, H.; Jung, J.; Kang, U. A Comparative Study of Matrix Factorization and Random Walk with Restart in Recommender Systems. In Proceedings of the 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, USA, 11–14 December 2017; IEEE: Piscataway, NJ, USA; pp. 756–765.
16. Guo, G.; Qiu, H.; Tan, Z.; Liu, Y.; Ma, J.; Wang, X. Resolving data sparsity by multi-type auxiliary implicit feedback for recommender systems. *Knowl.-Based Syst.* **2017**, *138*, 202–207. [\[CrossRef\]](#)
17. Margaritis, D.; Vassilakis, C.; Spiliotopoulos, D. What makes a review a reliable rating in recommender systems? *Inf. Process. Manag.* **2017**, *57*, 102304. [\[CrossRef\]](#)
18. Park, C.; Kim, D.H.; Oh, J.; Yu, H. Do “Also-Viewed” Products Help User Rating Prediction? In Proceedings of the 26th International Conference on World Wide Web, Perth, Australia, 3–7 April 2017.
19. Chen, C.; Zhang, M.; Liu, Y.; Ma, S. Neural Attentional Rating Regression with Review-Level Explanations. In Proceedings of the 2018 World Wide Web Conference, Lyon, France, 23–27 April 2018; pp. 1583–1592.
20. Mandal, S.; Maiti, A. Explicit Feedbacks Meet with Implicit Feedbacks: A Combined Approach for Recommendation System. In Proceedings of the Complex Networks and Their Applications VII. Complex Networks Studies in Computational Intelligence, Cambridge, United Kingdom, 11–13 December 2018; Aiello, L., Cherifi, C., Cherifi, H., Lambiotte, R., Lió, P., Rocha, L., Eds.; Springer: Cham, Switzerland, 2018. [\[CrossRef\]](#)
21. Mandal, S.; Maiti, A. Deep collaborative filtering with social promoter score-based user-item interaction: A new perspective in recommendation. *Appl. Intell.* **2021**, *51*, 7855–7880. [\[CrossRef\]](#)
22. Aciar, S.; Ochs, M. Classifying User Experience based on the Intention to Communicate. *IEEE Lat. Am. Trans.* **2020**, *18*, 1337–1344. [\[CrossRef\]](#)
23. Chen, R.; Hua, Q.; Chang, Y.S.; Wang, B.; Zhang, L.; Kong, X. A survey of collaborative filtering-based recommender systems: From traditional methods to hybrid methods based on social networks. *IEEE Access* **2018**, *6*, 64301–64320. [\[CrossRef\]](#)
24. Nikolakopoulos, A.N.; Ning, X.; Desrosiers, C.; Karypis, G. Trust your neighbors: A comprehensive survey of neighborhood-based methods for recommender systems. *arXiv* **2021**, arXiv:2109.04584.
25. Ren, L.; Wang, W. An SVM-based collaborative filtering approach for Top-N web services recommendation. *Future Gener. Comput. Syst.* **2018**, *78*, 531–543. [\[CrossRef\]](#)
26. Barathy, R.; Chitra, P. Applying Matrix Factorization in Collaborative Filtering Recommender Systems. In Proceedings of the 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 6–7 March 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 635–639.
27. Van der Vlugt, Y. Large-Scale SVD Algorithms for Latent Semantic Indexing, Recommender Systems and Image Processing. Bachelor’s Thesis, Delft University of Technology, Delft, The Netherlands, 2018.
28. Silveira, T.; Zhang, M.; Lin, X.; Liu, Y.; Ma, S. How good your recommender system is? A survey on evaluations in recommendation. *Int. J. Mach. Learn. Cybern.* **2019**, *10*, 813–831. [\[CrossRef\]](#)
29. Chen, M.; Liu, P. Performance evaluation of recommender systems. *Int. J. Perform. Eng.* **2017**, *13*, 1246. [\[CrossRef\]](#)
30. Karthikeyan, T.; Ravikumar, N. A Survey on Association Rule Mining. *Int. J. Adv. Res. Comput. Commun. Eng. (IJARCCE)* **2014**, *3*, 5223–5227.
31. Elyazgi, M.; Nilashi, M.; Ibrahim, O.; Rayhan, A.; Elyazgi, S. Evaluating the factors influencing e-book technology acceptance among school children using TOPSIS technique. *J. Soft Comput. Decis. Support Syst.* **2016**, *3*, 2.
32. Soto, D.L. Extensión al Modelo de Aceptación de Tecnología TAM, Para ser Aplicado a Sistemas Colaborativos, en el Contexto de Pequeñas y Medianas Empresas. Master’s Thesis, Universidad de Chile, Santiago, Chile, 2013.
33. Almenara, J.C.; Pérez, Ó.G.; Puente, Á.P.; Rosa, T.J. La Aceptación de la Tecnología de la Formación Virtual y su relación con la capacitación docente en docencia virtual. *Edmet. Rev. Educ. Mediát. TIC* **2018**, *7*, 225–241.
34. Prieto, J.C.S.; Migueláñez, S.O.; García-Peñalvo, F.J. Utilizarán los futuros docentes las tecnologías móviles? Validación de una propuesta de modelo TAM extendido. *Rev. Educ. Distancia* **2017**, *52*, 5. [\[CrossRef\]](#)
35. Jorge, G.A.; Mauro, P.S. Cálculo e interpretación del Alfa de Cronbach para el caso de validación de la consistencia interna de un cuestionario, con dos posibles escalas tipo Likert. *Rev. Publicando.* **2015**, *2*, 62–67.
36. Ulčar, M.; Robnik-Šikonja, M. High Quality ELMo Embeddings for Seven Less-Resourced Languages. In Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), Marseille, France, 11–16 May 2020; European Language Resources Association: Paris, France, 2020.

-
37. Athiwaratkun, B.; Wilson, A.G.; Anandkumar, A. Probabilistic Fasttext for Multi-Sense Word Embeddings. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, 15–18 July 2018; Association for Computational Linguistics (ACL): Stroudsburg, PA, USA, 2018.
 38. Tanaka, H.; Shinnou, H.; Cao, R.; Bai, J.; Ma, W. Document Classification by Word Embeddings of Bert. In Proceedings of the International Conference of the Pacific Association for Computational Linguistics, Hanoi, Vietnam, 11–13 October 2019; Springer: Singapore, 2019; pp. 145–154.