A Hybrid Multi-Criteria Hotel Recommender System Using Explicit and Implicit Feedbacks

Ashkan Ebadi, Adam Krzyzak

***Abstract*—**Recommender systems, also known as recommender engines, have become an important research area and are now being applied in various fields. In addition, the techniques behind the recommender systems have been improved over the time. In general, such systems help users to find their required products or services (e.g. books, music) through analyzing and aggregating other users' activities and behavior, mainly in form of reviews, and making the best recommendations. The recommendations can facilitate user's decision making process. Despite the wide literature on the topic, using multiple data sources of different types as the input has not been widely studied. Recommender systems can benefit from the high availability of digital data to collect the input data of different types which implicitly or explicitly help the system to improve its accuracy. Moreover, most of the existing research in this area is based on single rating measures in which a single rating is used to link users to items. This paper proposes a highly accurate hotel recommender system, implemented in various layers. Using multi-aspect rating system and benefitting from large-scale data of different types, the recommender system suggests hotels that are personalized and tailored for the given user. The system employs natural language processing and topic modelling techniques to assess the sentiment of the users' reviews and extract implicit features. The entire recommender engine contains multiple sub-systems, namely users clustering, matrix factorization module, and hybrid recommender system. Each sub-system contributes to the final composite set of recommendations through covering a specific aspect of the problem. The accuracy of the proposed recommender system has been tested intensively where the results confirm the high performance of the system.

***Keywords*—**Tourism, hotel recommender system, hybrid, implicit features.

I. Introduction

RECENT progress in information technology has provided us with various sources of data about almost everything. Although the availability of large-scale data can be beneficial, it can also make the decision making process more difficult. Users and customers have a lot of options to choose from which might make them confused in selecting the best possible and/or the most suitable item. In this sense, it is important to filter the information and personalize it for the use of each specific user. Recommender systems are one of the means for making personalized suggestions of items to the users based on their needs and preferences.

Nowadays, recommender systems are being widely used in different services covering vast area of applications. In parallel with the boost in tourism industry and data technology during the past decade, the travel recommender systems have attracted considerable attention of researchers. As a tourist, most of the times, it is really confusing to decide where to go and to select among a large number of possible destinations, especially for unseen and unfamiliar places [1]. Hence, information retrieval and decision support systems are widely recognized as valuable tools in this context. In this respect, tourism and travel recommender systems has become a hot topic recently and attracted the attention of both researchers and companies. However, most of the existing recommender systems in tourism employ a simple method which, in general, compares the profile of a given tourist with certain features of the available items (e.g. destination) and use them to predict the tourist's preferences [1], [2]. This is especially true about mobile recommender systems [3]. In such systems, a given tourist, i.e. the user, is asked to provide the system with a set of parameters that represent his/her interests, needs or limitations which are used by the system to make the recommendation through correlating the user's responses to the features of the available destinations/packages. These methods are also called as content-based recommendation [2].

Another approach is obtaining useful information from other tourists who have common or similar interests to the given user. In addition, systems can also benefit from the fact that travelers who are in close proximity might share common needs or interests [4]. Despite the recent advances in travel recommender systems, most existing recommender systems have been unsuccessful in exploiting the information, reviews, or ratings that are being provided by similar tourists [3].

In this paper, an intelligent hybrid hotel recommender solution is proposed. The proposed recommender engine is based on both the content data and similarities among users, exploiting implicit and explicit users' feedbacks. In addition, using data sources of different types, it employs multi-criteria rating approach to better capture users' preferences and augment the accuracy of the recommendations. The system is designed in different layers, using multiple sub-recommender systems each addressing specific aspect of the subject problem, aiming to increase efficiency and effectiveness of recommendations by considering. The proposed system is trained over TripAdvisor data, collected from multiple sources and integrated into a single database. The final solution is verified and tested in different settings and scenarios to confirm and validate its accuracy.

The rest of the paper is organized as follows: Section "Data

Ashkan Ebadi is with the Department of Computer Science and Software Engineering, Concordia University, Montreal, QC H3G 2W1 Canada (corresponding author, phone: 352 745-4468; e-mail: a_ebad@encs.concordia.ca).

Adam Krzyzak is with the Department of Computer Science and Software Engineering, Concordia University, Montreal, QC H3G 2W1 Canada (e-mail: krzyzak@cs.concordia.ca).

and Methodology" describes methodology and data that will be used in this study. The experimental results, performance evaluations, and interpretations are provided in section "Results". Conclusions are made in section "Conclusion", and limitations along with some directions for the future work are discussed in the last section "Limitations and Future Work".

II. Data and Methodology

Two separate datasets were collected and used: 1) For training the sentiment analysis module, and 2) For training the recommender system as well as performing automatic keyword extraction. Machine learning methods typically require large amounts of data to provide sufficient predictive power. Since no labelled large-scale travel-specific training corpuses exist to this date, we used other *user expression* resources. In particular, *Twitter* was selected as the source of the training data for the sentiment analysis module and a corpus of 1.6 million tweets[1], labelled as positive and negative, was collected from the Internet and used as the training dataset. Since people mainly use quick and short messages and due to the automatic collection of the data, the data was not clean hence we first preprocessed and cleaned the collected Tweets dataset by removing the hashtags[2] and hyperlinks from the tweets, and removing the target users[3] from the twitter data and trimming the texts.

TripAdvisor was mainly used as the data source for training the recommender systems and performing the keyword extraction. TripAdvisor employs user-generated content. The TripAdvisor website is free to use and the company's business plan is based on the support from advertisement. We collected the TripAdvisor travel data from multiple sources since we wanted to have a complete corpus of user reviews, hotel ratings (multi-aspect), as well as complete hotel information. For this purpose, we used Li, Ritter and Hovy [6] data which contain a complete list of hotels including all the respective information about the hotels such as class, region, physical address, website, *etc.* as well as users' reviews on the listed hotels. This comprehensive dataset, which was used in a number of research papers, contained 878,561 users' reviews on 4,333 different hotels, about 1.3 Gigabytes, which was crawled from the TripAdvisor website. In addition to the mentioned dataset, we also used another dataset which contains 246,400 hotel reviews [7]. These data contain numerical ratings, ranging from 1 to 5, provided by users on different aspects of the hotels, *e.g.* value, room quality, location, and service, along with other complementary information about the hotels. Textual users' reviews on hotels are also available. The data were preprocessed by removing all the excess spaces and tabs and converting commas to semi-colons.

Having collected the required hotel data, we integrated the collected data into a MySQL database. For this purpose, different entities, *e.g.* hotels and users, were first identified and their unique IDs were considered as the primary key in respective tables. Next, an automatic data integration procedure was coded in JAVA which went through all the records and integrated them into the database, through checking for duplicates and inserting all the related information about an entity in the respective row. The final database contains 4,333 distinct hotels with complete information about them. 148,429 users have rated 1,850 different hotels focusing on various aspects. In addition, 148,421 users have written text reviews about the hotels in the integrated dataset. We performed further preprocessing on the integrated data including noise removal, *i.e.* removing the rows with so many missing values, hyperlink removal, performing spell check on the collected data, and converting all the words into lower cases.

Having all the required data collected, we employed a number of different tools, methods and methodologies for designing and implementing the hybrid hotel recommender system. In general, machine learning techniques and natural language processing (NLP) were used to develop the core of the recommender engine. The entire recommender system contains three different sub-systems, *i.e.* sentiment analysis module, keyword extraction module, and the recommender engine that are presented separately in this section.

A. Sentiment Analysis Module

This module automatically detects the positivity and negativity of users' text reviews and provides the recommender engine with a complementary implicit feature, in terms of a polarity score, reflecting user's satisfaction. We applied machine learning and text processing techniques to provide an automated mechanism for detecting the sentiments of users' reviews and creating an implicit polarity score. The Tweets database was used to train and develop the sentiment analysis model. For this purpose, a text mining engine was designed which takes the Tweets as the input and converts them to numerical form, suitable for training the sentiment classifier. The input text was already converted to lower case at the time of integrating the travel data. The punctuations were also replaced with a blank space. Next, the streams of text were broken into the smallest meaningful components, called *tokens*. We then converted the tokens to *stems* by removing and replacing the word suffixes to obtain the common root of the word.

We considered both unigrams and bigrams[4] of the tokens. In order to obtain numerical feature vectors, all the considered tokens were then converted to count vectors in which the index value of a token was linked to its frequency in the entire training corpus. To normalize the feature vector, we employed the term frequency-inverse document frequency (*tf-idf*) approach. After preprocessing the text data, a logistic regression model was built using 10-fold cross validation approach to assure the accuracy of the learned model.

The generated sentiment analysis model was found to be 85% accurate in predicting the polarity of Tweets that means

[1] The original dataset was taken from [5].

[2] Users of Twitter usually use hashtags to refer to or mark topics.

[3] Users of Twitter use the "@" symbol to refer to other users.

[4] In general, an *n-gram* is a contiguous sequence of *n* items (tokens or stems) from a given sequence of text.

the model has promising performance in predicting the sentiment of a given sentence/text, comparing to the literature. For example, Pak and Paroubek [8] achieved 81% of accuracy in predicting positivity and negativity of tweets. In a recent survey study, Rosenthal et al. [9] listed the performance results of 11 different systems in predicting phrase-level binary polarity of tweets, where all the systems have accuracy lower than 85%. Using the sentiment model, the polarity of the given user's review is predicted and an intermediate output, namely the *polarity score* is generated. The system reports a value in the range of [1], [5] as the polarity score where the score reflects the intensity of polarity: closer to 1 value represent more negative sentiment, and closer to 5 value indicate more positive sentiment. This polarity score is used as an implicit feature in the recommender engine

B. Keyword Extraction Module

Summarizing and analyzing the content of the users' reviews can be very beneficial. The main objective of the keyword extraction sub-system is to go through the users' reviews and extract keywords out of them, and assign them as implicit features to the respective users. We employed Latent Dirichlet Allocation (LDA) method to extract the topics out of the set of reviews. The extracted topics were then refined to find the representative set of keywords for each user in the database. The refinement stage was added to the sub-system as LDA might result in soft clusters where a semi-automatic refinement procedure can filter the undesired results and improve the accuracy. The whole keyword extraction procedure is depicted in Fig. 1. As seen, users' reviews are first collected and preprocessed. In particular, we removed special characters from the data as they could affect the accuracy of the system negatively. In addition, reviews that were in a language other than English (*e.g.* Chinese, French) were removed. Next, we removed the English stop words[5] from the vocabulary and numerical values from the data as they were not informative in the defined keyword extraction task. The clean users' reviews dataset was then split into a set of reviews for each user. Next, LDA was performed on each set of reviews separately, and the set of specific keywords for each user are extracted. Finally, a semi-automatic procedure refined the extracted keywords. The final extracted keywords were assigned to the users as implicit features which partially reflected their interests.

C. The Recommender Engine

The recommender engine itself consists of three major modules: 1) *User Clustering Module*, 2) *Matrix Factorization Module*, and 3) *Hybrid Recommender Module*, as shown in Fig. 2. The Users Clustering Module clusters all the users in the system into different groups based on their characteristics. The distance of any given user's features set is then compared with the centroids of the generated clusters, and the best cluster is selected for the user. Next, the Matrix Factorization and Hybrid Recommender Module are provided with the selected cluster, separately, where the final recommendation is made based on the outputs of the two mentioned modules.

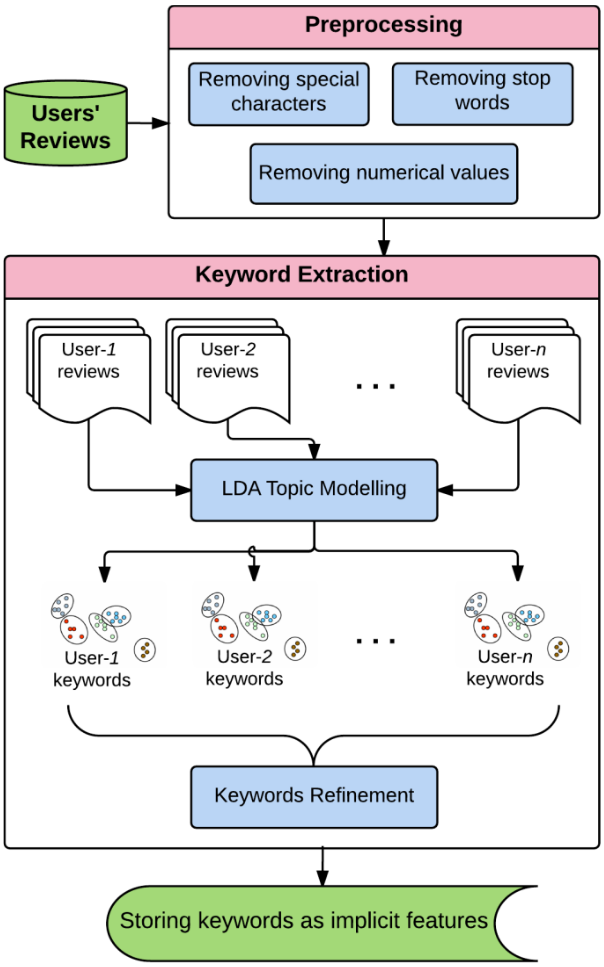

Fig. 1 The keyword extraction module. Users' reviews are first preprocessed. The preprocessed data are then used in the keyword extraction module where machine learning LDA topic modelling technique is used for extracting the keywords for each user based on their reviews. The detected keywords list is then manually refined and the final implicit features are stored

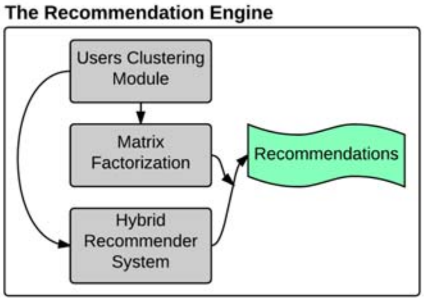

Fig. 2 The recommender engine contains three main modules, namely user clustering, matrix factorization, and the hybrid recommender system. Users are first clustered based on various features. The selected cluster is then fed into the matrix factorization module and the hybrid recommender system. The output of the mentioned two modules forms the final set of recommendations

[5] Stop words are the most common words which are in the text but can be of little value in detecting the most informative keywords.

Clustering module was implemented and included, mainly due to the high sparsity of the users' data features. Clustering techniques can reduce the sparsity and improve the performance and scalability of the recommender systems [10], [11]. The Users Clustering Module employs two different set of data features to do the clustering, namely users' demographics and the detected keywords. Users' demographics contain information such as user's age, gender, date of registration in the system, location, *etc*. To increase the data dimension, we also included the detected user-specific keywords, obtained from the Keyword Extraction Module, as explained in the previous section.

To perform the clustering, *K-Means* clustering approach [12] was first selected. The ease of implementation, speed, and the fact that *K-Means* performs reasonably well in large datasets [13], were the main reasons for such selection. However, since the data contained both categorical and numerical variables (features), we used a variation of *K-Means*, named *K-Prototypes* [13], which is able to handle both categorical and numerical features. The number of clusters/prototypes (*k*) should be given to the K-Prototypes algorithm. Thus, finding the optimal number of clusters, the best *k*, was crucial and could affect the performance of the system. We used the *Gap statistic* [14] for estimating the best *k* for the users' data. We found that the Gap statistic peaks at *k* = 4 with the value of ~1.006. Thus, the existence of 4 clusters was confirmed. Therefore, using the K-Prototypes algorithm, users were clustered into 4 different groups.

In recommender systems, the *cold start* is a well-known problem. This refers to the fact that the recommender system is not able to provide any recommendation for users or items that have not enough information about. This might cause the system to make common and/or the same recommendations to the users. Specifically, in the case of collaborative filtering, the system works by identifying the users with similar/same preferences to the given user, and then recommends the items that those users, but not the given user, have already favored. Thus, it will fail to suggest items for which there exist no ratings, *e.g.* new items to the community [15]. We used non-negative matrix factorization technique [16] to stand for this problem in our proposed recommender system. NMF has been widely used in various fields and applications, including recommender systems [17], [18]. Using NMF, the user-item ratings matrix was factorized into two matrices, such that all the three matrices include only non-negative elements. This module particularly addresses the cold start problem through providing recommendations on unseen hotels to the users.

The final (and main) part of the proposed recommender system is the Hybrid Recommender Module. The main goal of this module is to leverage from different types of features to build a hybrid model. It employs a set of features which reflect users and hotels characteristics. These features are called as explicit features. In addition, a complementary set of implicit features, *e.g.* users' reviews polarity, are also used in the model. The sentiment analysis module, and the tf-idf approach in particular, along with the keyword extraction module, enable the hybrid recommender engine to exploit the content, thus, providing more dimension to the system. The overall architecture of the module is depicted in Fig. 3. First the best cluster is detected for a user based on various features and characteristics, and then the hybrid recommender system is provided with the selected user cluster. The hybrid recommender system consists of three sub-modules: 1) User-based collaborative filtering module, 2) Item-based collaborative filtering module, and 3) Multi-criteria recommender system.

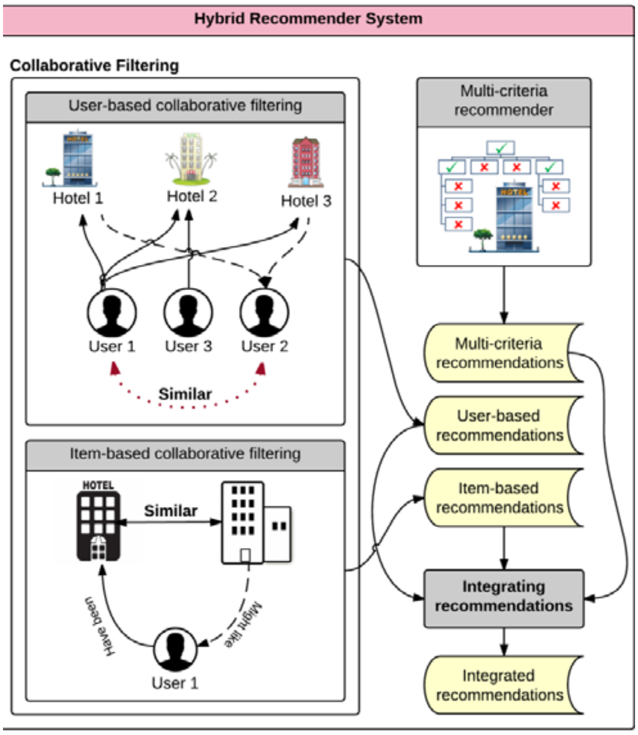
Hybrid Recommender System
Collaborative Filtering
User-based collaborative filtering
Hotel 1
Hotel 2
Hotel 3
User 1
User 3
User 2
Similar
Item-based collaborative filtering
HOTEL
Similar
Have been
Might like
User 1
Multi-criteria recommender
Multi-criteria recommendations
User-based recommendations
Item-based recommendations
Integrating recommendations
Integrated recommendations

Fig. 3 The hybrid recommender module itself contains of three main sub-modules: 1) user-based collaborative filtering module, 2) item-based collaborative filtering module, and 3) multi-criteria recommender. The system produces four different outputs, i.e. recommendations made by each of the mentioned sub-modules along with an integrated composite set of recommendations which is made by combining all the three previously stated recommendations

The user-based collaborative filtering module makes recommendation based on the similarities among users. To measure the similarities among users, various similarity measures were tested. The best and most robust result was found for *cosine similarity*. Thus, the cosine similarity was used in the user-based collaborative filtering module for calculating the similarity among users. Using the cosine similarity measure, this module makes recommendations in two steps: 1) Identifying users who share the same rating pattern with the given user, and 2) Using the ratings from the similar users who were found in the first step for predicting the ratings for the given user. The rating is only predicted for the hotels that the given user has not already been there. The hotels (items) are sorted based on their score, and the top hotels in the sorted list are recommended.

The item-based collaborative filtering module checks the similarities among various hotels. Similarities are calculated based on the cosine similarity measure. Using the calculated similarities among the hotels (items), rating predictions are made for <user, hotel> pairs that are not present in the dataset, *i.e.* this module recommends hotel to a user that have not already been seen by him/her. Similarities between two given hotels are calculated using all the users who have rated both the hotels. After modelling the data using the cosine similarity measure, the weighted sum approach is taken for predicting the rating for any (unseen) <user, hotel> pair. That is, all the hotels that are similar to the candidate hotel are first selected, forming the set of *similar hotels*. From the similar hotels, the algorithm selects the ones that have been already rated by the given user. The user's ratings for each of these found hotels are weighted, using the similarity between that hotel and the candidate hotel. Finally, the predictions are scaled by the sum of similarities, and the top hotels are recommended. Item-based filtering module is considerably faster than the user-based collaborative filtering module, however, the item similarity table should be maintained and updated over time.

To overcome the limitation of the single criterion value, *i.e.* the overall rating, a multi-criteria recommendation engine has been also implemented. This module contributes to the hybrid recommender system through improving the quality of the final recommendations by representing more complex preferences of each user, as the suitability of the recommended hotel for a given user might depend on more than one rating aspect. The multi-criteria recommender module is provided with multi-criteria rating data, *i.e.* users rated on multiple and different aspects of the hotels. The additional information on each user's preferences would help to improve the model accuracy and capability in learning users' preferences. We designed this module as a user-based multi-criteria collaborative filtering approach. The architecture is almost the same as the user-based collaborative filtering, as explained before. The only difference is in defining and calculating the similarity measure in order to stand for the availability of multi-criteria information. There exist a number of studies that focused on extending the traditional similarity measure calculations to reflect multi-criteria information [19], [20]. One common approach is to aggregate the traditional single-criteria similarities. We modified the cosine similarity to reflect the multi-criteria information. In particular, the similarity between any two given users was calculated based on each individual criterion, let us say k criterions, using the single criterion cosine similarity measure. Then, the final similarity between the two given users was calculated by averaging the calculated k similarities [19]. This aggregated users' similarity measure was then used by a user-based collaborative filtering module to make the recommendations.

Each of the sub-modules of the hybrid recommender engine produces a separate set of recommendation, addressing a specific aspect of the problem. In addition to these three recommendation set, a composite set of recommendations is also made, using all the outputs of the internal modules.

III. Performance Evaluation

We did several experimental evaluations to assess the performance of the proposed recommender system. The leave-one-out cross validation (LOOCV) approach was selected for validating the results. In LOOCV with n data points, 1 observation (data point) is considered as the validation set in each run, while the remaining data points form the training set. The procedure is repeated n times, taking all data points as the validation set once. A set of decision-based error measures, *i.e.* accuracy, specificity, sensitivity, and informedness, as well as three prediction-based error metrics, *i.e.* mean absolute error (MAE), mean squared error (MSE), and root-mean-square error (RMSE), were calculated for evaluating the proposed recommendation system. In the rest of this section, the performance evaluation results including the evaluation of all the sub-modules as well as the composite set of recommendations, are presented and discussed in detail.

A. Matrix Factorization Module

The matrix factorization module approximates the missing rating values. The module was designed as part of the hybrid recommender system to account for the new to the system hotels, as they cannot be perfectly identified and recommended by collaborative filtering based approaches. Since the user-hotel rating matrix was very sparse, the nonnegative double singular value decomposition (NNDSVD) approach was used for initialization. NNDSVD which is method for enhancing the initialization stage of the NMF approach is proven to be very effective for rapid reduction of NMF algorithm estimation error [21].

As seen in Fig. 4, RMSE of the NMF module is considerably small, with the value of 0.364, and is larger than MAE and MSE as expected, according to the definition of the mentioned measures. Based on the observed measures, *i.e.* MAE, MSE, and RMSE, it can be said that the NMF module is highly accurate in decomposing the user-hotel rating matrix into two matrices and predicting the ratings for any user-hotel pair. From the results it can be said that NMF module can be employed with high accuracy to recommend (unseen to the user or new to the system) hotels to the users.

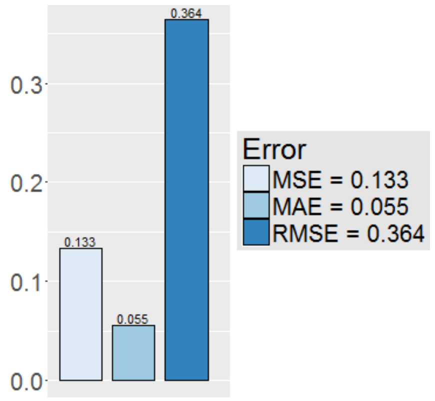

Fig. 4 Prediction-based metrics calculated for NMF module

B. User-Based Collaborative Filtering Module

Fig. 5 depicts the prediction-based metric for the user-based collaborative filtering module. As seen, the user-bases CF is able to predict the ratings for user-hotel pairs with relatively small error. According to MAE measure, the average magnitude of the errors in the prediction set, regardless of their direction, is relatively small, with the value of 0.65. This can be also regarded as a sign of high accuracy of the user-based CF sub-system. In other words, the average of the absolute values of differences between the predicted ratings and the corresponding observations over the entire LOOCV validation procedure is slightly higher than half a scale, indicating the high performance of the sub-system. One should note that the error rate is promising considering the large data size that was used.

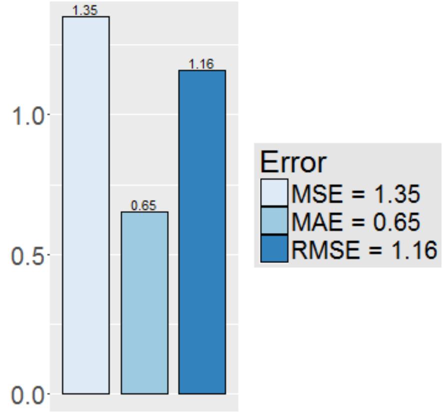

Fig. 5 Prediction-based error metrics calculated for the user-based collaborative filtering module

MSE and RMSE are also positive numbers, ranging from 0 to ∞, where lower values indicate higher accuracy. RMSE is of special interest, as large rating prediction errors are not desirable since they might lead to wrong recommendations. According to the results, RMSE of the user-based CF module is also relatively small, with the value of 1.16. The difference between RMSE and MAE is approximately equal to 0.5, *i.e.* 1.16 – 0.65 = 0.51, which indicates that the variance in the individual errors is also relatively small. From Fig. 5, it can be observed that the mean square error of the user-based CF module is relatively small, as well.

As seen in Fig. 6, the module is more than 83% accurate in predicting the right hotels to be recommended to the users. This confirms that the module is able to make robust, accurate, and acceptable predictions through learning the users' preferences accurately. According to the results, the user-based module is highly sensitive, *i.e.* 94.1%, indicating that the model is highly complete, capturing almost all the positives in the data. Thus, the model is able to effectively learn users' preferences. Specificity of the model is also considerably high (83.5%). Specificity can be regarded as the effectiveness of the system in identifying true negatives. From the calculated specificity and sensitivity of the model, it is clear that the model is able to learn the preferences, correctly identifying both desirable and undesirable items. This is of high importance in travel recommendation systems, as unintelligent recommendations might cause users to even leave the system. Moreover, from the informedness measure (77.7%), it can be said that the module is appropriately using the information hidden in the data to make informed decisions in recommending hotels to similar users in the system.

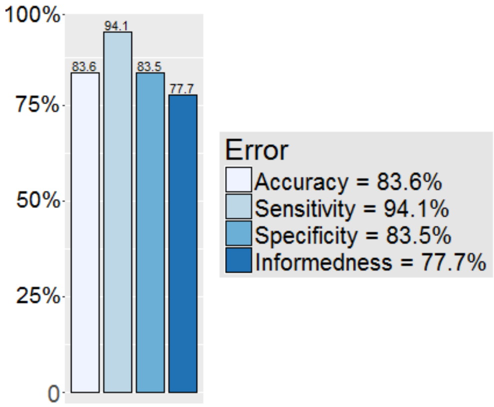

Fig. 6 Decision-based performance metrics, calculated for the user-based collaborative filtering module

C. Item-Based Collaborative Filtering Module

This module accounts for the question of "which hotels are similar to each other?". Item-based collaborative filtering helps the proposed hotel recommendation systems, at least, in two ways: 1) By recommending items (hotels) to users that have not already been rated by them, thus, solving the problem of new items to users, and 2) By improving the overall speed of the system and the possibility of acting as a fast independent recommender module, if necessary. As discussed in section II, the item-based collaborative filtering module was designed such that it recommends unseen hotels that are similar to the ones that have been already rated by a user. That is, we included this module to act as a complementary component to the other modules in the recommender engine, recommending unseen hotels. Therefore, it is not possible to calculate the accuracy of this sub-module, using the current setting. However, in operation, several strategies can be taken by the operating website to investigate if users have welcomed the new hotel suggestions. This can be done, for example, by incorporating web cookies in order to capture user navigation traces and behaviors.

D. Multi-Criteria Recommender Module

We checked the performance of MCR module for the case that implicit and explicit feedbacks are used to assess different aspects of the hotels and form the rating vectors for user-hotel pairs. Same as the other modules, LOOCV was used. As seen in Fig. 7, MAE metric that is a measure for the average magnitude of the errors in the predictions set without considering their directions, is considerably low. Since the MSE and RMSE scores are also significantly low, it can be said that the module is highly accurate in predicting the ratings for user-hotel pairs. Comparing the results with Fig. 5, it is observed that the multi-criteria recommender system is performing better than the user-based collaborative filtering module. Although MAE of the multi-criteria recommender is slightly higher, i.e. 0.76 vs. 0.65, since the other metrics are much better for the multi-criteria system, it can be said that the multi-criteria recommender is able to benefit from the multi-aspect evaluations to enhance the quality of the recommendations. Moreover, the very small difference between MAE and RMSE in the multi-criteria recommender indicates that the variance in the individual errors is considerably small, even better than the user-based collaborative filtering system. That is, the model is making relatively small errors in predictions.

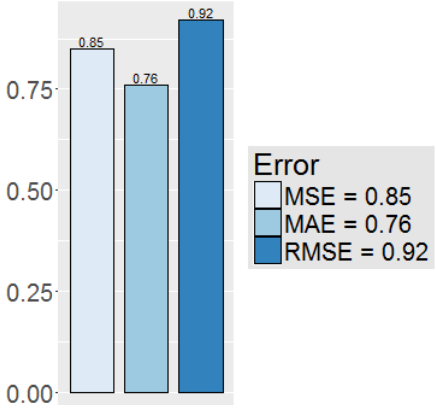

Fig. 7 Prediction-based metrics for the MCR module, using implicit and explicit ratings

Analyzing the decision-based metrics (Fig. 8) revealed that the module is more than 90% accurate in predicting the right hotels to recommend to the users. This confirms that the module is able to make robust, accurate, and acceptable predictions. As expected, this is higher than the accuracy of the user-based collaborative filtering module (Fig. 6). Analyzing the other measures also confirms that the multi-criteria recommender model is well fitted for the subject problem. That means, the multi-criteria recommender module is very sensitive, *i.e.* 91%, indicating that the model is highly complete, capturing almost all the positives in the data. Thus, the model is able to effectively and (almost) completely learn users' preferences. Moreover, the specificity of the model is also significantly high, exceeding 90%. Therefore, the module can effectively identify true negatives. According to the specificity and sensitivity metrics, the model is highly capable of learning users' preferences such that desirable and undesirable items are correctly identified. Finally, the informedness measure (81.1%) shows that the module is appropriately detecting the hidden information in the data and the patterns of preferences, and employs them to make informed decisions in recommending hotels to similar users in the system.

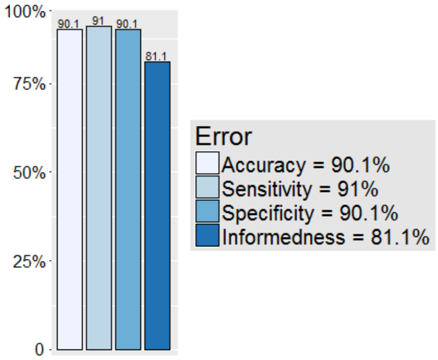

Fig. 8 Decision-based performance metrics, calculated for the multi-criteria recommender module

E. The Composite Set of Recommendations

We integrated the recommendations from the previously discussed sub-modules, giving higher weight to the multi-criteria recommender due to higher performance, and formed the composite set of recommendations. The predicted ratings were compared with the actual *total* ratings. One should note That is, the total rating, which is the single rating that a user gives to a hotel, was considered as the ground truth. The results of the prediction-based metrics calculations for the composite set of recommendations revealed that the system performs reasonably well in predicting the ratings for user-hotel pairs (Fig. 9).

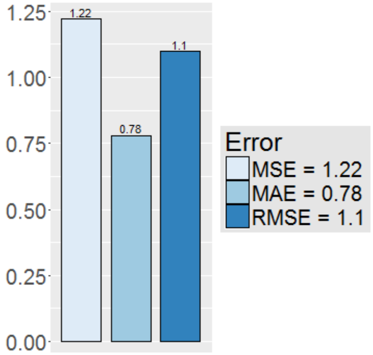

Fig. 9 Prediction-based metrics calculated for the composite set of recommendations

The relatively small difference between RMSE and MAE confirms the existence of low variance in the individual errors. While indicating there is some variation in the magnitude of errors, it also confirms that large errors are very unlikely. According to the results, the average difference between the predicted rating based on the composite set of recommendations and the observed total rating is 0.78. This highlights the power of the proposed system in learning the users' preferences. Interestingly, the accuracy of the composite set of recommendations also exceeds 98%.

IV. Conclusion

In this paper, a novel hybrid solution was proposed for predicting ratings for user-hotel pairs and making the recommendation. The proposed approach combined collaborative filtering with matrix factorization and clustering techniques to improve the performance. Moreover, users' text reviews were converted to polarity scores, reflecting implicit feedbacks, and were integrated into the feature space. In addition, topic modelling techniques were applied to generate implicit features from users' reviews, reflecting unique points of interests for each user in the system. The diversity of the features types, including both implicit and explicit feedbacks, as well as the integrity of the techniques, helped the system to reach outstanding accuracy and performance. Although there are hybrid recommendation designs in the literature, to the best of my knowledge, this is the first one in hotel recommendation domain that applies a triangulation technique and incorporates sentiment analysis and keyword extraction techniques to obtain content information and use them along with a diverse set of other features to solve the problem. The main advantages of the proposed design are: 1) A hybrid design which is well suited to the subject problem and can be operated easily, 2) Highly accurate predictions, 3) Use of implicit and explicit feedbacks and the novelty in employing sentiment analysis and keyword extraction techniques for extracting new features, 4) The system's ability in recommending "new to the user" items as well as "unseen" ones, and 5) Benefitting from a multi-criteria rating system that helped the recommender engine to better learn users' preferences.

To speed up the system as well as to improve its accuracy, clustering techniques were applied on users' vectors, grouping them in various clusters based on their characteristics. This also played an important role in improving the recommender system performance in comparison with the basic collaborative filtering algorithms. For this purpose, the system employed various types of content features such as user's age, and location. Moreover, matrix decomposition techniques were employed to solve the cold start problem, making the system capable of drawing inferences for the new to the system items about which it has not yet enough information. The multi-criteria recommender module also empowered the system with multi-aspect ratings that enabled it to provide more accurate recommendations. In addition, this thesis presents an innovative technique for extracting implicit features and converting them to an implicit rating score. The use of implicit features here was found to be crucial, as it was observed that incorporating them augments the system performance through providing it with deeper understanding of user preferences and characteristics.

Comparing the results with the literature, it was observed that the sentiment analysis module shows promising performance in predicting the sentiment of a given sentence/text, with 85% accuracy. This is higher than several similar studies such as [8], [9][6]. The proposed recommender framework also performs more effectively than the approaches in the literature. For example, [22] proposed a multi-criteria recommender system for tourism domain and compared the results with several other algorithms, using TripAdvisor data. According to their findings, mean absolute error for the standard collaborating filtering method [19] equals 1.37. MAE was found to be 1.28 for Total-Reg algorithm [19], and, 0.89 for ANFIS and HOSVD algorithm [11]. As seen, the recommender engine that was proposed in this thesis outperforms the similar available systems.

In general, it can be said that the proposed solution can fit well with the hotel recommendation problem, covering all the aspects that might be important in a real-life business case. It is also flexible enough to take the speed-accuracy trade-off into the account through giving higher weights to different sub-systems based on the available conditions in the company and the market situation. The system is also highly customizable and can be easily adjusted for different scenarios or even different businesses, with some minor changes. However, apart from the system architecture, the error metrics are also required to be selected wisely, in accordance with the business nature and problem objective(s).

V. Limitations and Future Work

Recommender systems are now being used widely in various types of applications and domains. There are a number of traditional ways to measure recommenders' architecture and performance. However, a precise evaluation of a recommender system is more accessible in an actual situation. Such situation should be properly evaluated providing a clear picture of the domain properties and characteristics, business goals and objectives, and behavior of the algorithm. The operational data such as time that a user spent on a web page, number of hits for recommendations, click tracking, like or dislike for a recommendation, *etc.*, can contribute to better evaluation of the proposed recommender.

Another direction for the future research might be using more data. Although large scale data were used in this paper for training the recommender modules, more data (from other sources) can be definitely helpful. In addition, more features on users and/or hotels can surely help the system, at least in performing a better clustering of users which might ultimately lead to higher performance. Moreover, the user-hotel-rating matrix is extremely sparse, thus, testing the proposed system on less sparse data can be also suggested. Although multi-aspect rating was also used in this system, more rating and/or more rating dimension can be helpful in determining the behavior of the hybrid recommender engine more accurately.

References

[1] F. Ricci, "Travel recommender systems." *IEEE Intelligent Systems*17.6, pp. 55-57, 2002.

[2] D. Gavalas and M. Kenteris, "A web-based pervasive recommendation system for mobile tourist guides." *Personal and Ubiquitous Computing* 15.7, pp. 759-770, 2011.

[3] W. S. Yang and S. Y. Hwang, "iTravel: A recommender system in mobile peer-to-peer environment." *Journal of Systems and Software* 86.1, pp. 12-20, 2013.

[4] A. de Spindler, M. C. Norrie, M. Grossniklaus and B. Signer, "Spatio-temporal proximity as a basis for collaborative filtering in mobile environments.", 2006.

[6] In this survey, performance results of 11 different systems in predicting phrase-level binary polarity of tweets were listed, where all the systems have accuracy lower than 85%.

[5] A. Go, R. Bhayani and L. Huang, "Twitter sentiment classification using distant supervision." *CS224N Project Report, Stanford* 1: 12, 2009.

[6] J. Li, A. Ritter and E. H. Hovy, "Weakly Supervised User Profile Extraction from Twitter." In *ACL (1)*, pp. 165-174, 2014.

[7] H. Wang, Y. Lu, and C. Zhai, "Latent aspect rating analysis without aspect keyword supervision." In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 618-626. ACM, 2011.

[8] A. Pak and P. Paroubek, "Twitter based system: Using Twitter for disambiguating sentiment ambiguous adjectives." In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pp. 436-439. Association for Computational Linguistics, 2010.

[9] S. Rosenthal, P. Nakov, S. Kiritchenko, S. M. Mohammad, A. Ritter and V. Stoyanov, "Semeval-2015 task 10: Sentiment analysis in twitter." *Proceedings of SemEval-2015,* 2015.

[10] A. Bilge and H. Polat, "A scalable privacy-preserving recommendation scheme via bisecting k-means clustering." *Information Processing & Management* 49, no. 4, pp. 912-927, 2013.

[11] M. Nilashi, O. bin Ibrahim and N. Ithnin, "Multi-criteria collaborative filtering with high accuracy using higher order singular value decomposition and Neuro-Fuzzy system." *Knowledge-Based Systems* 60, pp. 82-101, 2014.

[12] J. MacQueen, "Some methods for classification and analysis of multivariate observations." In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, no. 14, pp. 281-297, 1967.

[13] Z. Huang, "Clustering large data sets with mixed numeric and categorical values." In *Proceedings of the 1st Pacific-Asia Conference on Knowledge Discovery and Data Mining, (PAKDD)*, pp. 21-34, 1997.

[14] R. Tibshirani, G. Walther and T. Hastie, "Estimating the number of clusters in a data set via the gap statistic." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 63, no. 2, pp. 411-423, 2001.

[15] A. I. Schein, A. Popescul, L. H. Ungar and D. M. Pennock, "Methods and metrics for cold-start recommendations." In*Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 253-260. ACM, 2002.

[16] S. Sra and I. S. Dhillon, "Generalized nonnegative matrix approximations with Bregman divergences." In *Advances in neural information processing systems*, pp. 283-290, 2005.

[17] R. Gemulla, E. Nijkamp, P. J. Haas and Y. Sismanis, "Large-scale matrix factorization with distributed stochastic gradient descent." In*Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 69-77. ACM, 2011.

[18] Y. Bao, H. Fang and J. Zhang, "TopicMF: Simultaneously Exploiting Ratings and Reviews for Recommendation." In *AAAI*, pp. 2-8, 2014.

[19] G. Adomavicius and Y. O. Kwon, "New recommendation techniques for multicriteria rating systems." *Intelligent Systems, IEEE* 22, no. 3, pp. 48-55, 2007.

[20] N. Manouselis and C. Costopoulou, "Experimental analysis of design choices in multiattribute utility collaborative filtering." *International Journal of Pattern Recognition and Artificial Intelligence* 21, no. 02, pp. 311-331, 2007.

[21] C. Boutsidis and E. Gallopoulos, "SVD based initialization: A head start for nonnegative matrix factorization." *Pattern Recognition* 41, no. 4, pp. 1350-1362, 2008.

[22] M. Nilashi, O. bin Ibrahim, N. Ithnin and N. H. Sarmin, "A multi-criteria collaborative filtering recommender system for the tourism domain using Expectation Maximization (EM) and PCA–ANFIS." *Electronic Commerce Research and Applications* 14, no. 6, pp. 542-562, 2015.