Case-based Mobile Tourism Attractions Recommender System

Septia Rani¹, Dimastyo Muhaimin Arifin², Sheila Nurul Huda³, Dhomus Hatta Fudholi⁴

¹³⁴Department of Informatics, Universitas Islam Indonesia
²Undergraduate Program in Informatics, Universitas Islam Indonesia

E-mail: ¹septia.rani@uii.ac.id, ²dimastyo.arifin@students.uii.ac.id

Abstract. The variations and the increasing number of tourism attractions give tourists more choices to enrich their travel experience. However, it also presents drawbacks, including confusing tourists when planning a trip, because they have to choose from many tourist attractions options that match their preferences. In this study, we propose an algorithm that can generate recommendations of tourist attractions to the user using a case-based reasoning approach. Case-based recommendation is chosen because it gives the user a more personalized recommendation that they can customize later on within the application. Besides, the implementation of this approach in the tourism domain is still hard to find. As many as 217 tourist attraction objects in Yogyakarta city are gathered as the case study. We label each object with three attributes, namely category, visitor type, and activities. A similarity function called the simple matching coefficient is used to calculate the item’s score based on the user specification. For assessing the proposed algorithm, we implement the model as an Android application. To evaluate the recommendation result, we use mean absolute error calculation (MAE) and mean absolute percentage error (MAPE), resulting in 4.1 MAE and 5% MAPE, which is considered highly accurate. Thus, the recommendation from the system is well suited to the user’s preference.

Keywords: case-based reasoning, recommender system, tourist attraction.

1. Introduction

In the past few years, the tourism sector has received the attention of many people. According to the World Tourism Organization (UNWTO), tourism comprises the activities of persons traveling to and staying in places outside their usual environment for not more than one consecutive year for leisure, business, and other purposes not related to the exercise of an activity remunerated from within the place visited [1]. One of the factors that influence the tourism sector's progress is the innovation that causes the emergence of many variations in tourism destinations, tourism products, business models, and services.

The variations and the increasing number of tourism destinations bring advantages for both tourists and tourism providers/operators. However, sometimes the vast offer of options presents its drawback for future travelers. To gain the best experience during their travel, travelers must possess a good judgment or understanding of their potential destination. Travelers who do not have this trait may find...
it hard to decide their destination, given their lack of information. It becomes clear that this issue must be addressed.

This study aims to build a mobile application that recommends tourist attractions for future travelers. We propose case-based reasoning as the algorithm to generate a recommendation. In a case-based recommender system, specific cases are specified by the user as target points, and the similarity metrics are used to retrieve similar items to these cases [2]. During the registration step in our proposed system, users are asked to fill their profile and choose several samples of tourist attractions that they like. After that, each time users want to plan their trip and get a recommendation, they must specify their specifications. The user specification (known as the case) is then used as the baseline and combined with domain knowledge to recommend tourist attractions. One of the advantages of using a case-based recommendation is that these selected cases can be altered anytime by the user through their profile so that they may receive different recommendations if they are not satisfied by the current recommendation.

The rest of this paper is organized as follows. Section 2 describes related work. Section 3 describes the methodology which is being used in this study. Section 4 describes the result and discussion. Finally, section 5 concludes the overall remarks of this study.

2. Related Work
Recommender system is an area with many possibilities of application, rise to fame in the last decade along with many flourishing topics in the Artificial Intelligence field. Recommender system for tourism is one of many interesting applications, especially since the tourism sector is considered one of the biggest sectors globally [3]. A tourism recommender system can help the user get a personalized tourist attraction recommendation for a city that they have never visited before. Over the last decade, research efforts on the recommender system for tourism have been conducted by many researchers with a diverse and interesting idea. The most common categorization of the recommender system is content-based filtering, which recommends similar items and collaborative filtering that recommend items from similar people. There is also knowledge-based filtering [4]. Some add another touch to make the recommendation more interesting, such as serendipitous recommendation [5] or context-aware recommendation [6] to address specific issues.

Recommendation (or decision) making can sometimes be a challenging task, as the basis of recommendation is data which can be too scarce. For example, in a cold start application with no previous user preferences data or too much data (for example, an application that collects thousands of rating data from thousands of users). Among the researches which use rating data from the user to generate recommendation are [7, 8], known as collaborative filtering. The problem in collaborative filtering lies in the availability of sufficient and adequate number of data. Another approach is needed for a cold start application with no (or limited number) of user data.

Another approach using the case-based method on tourist recommendation falls under the knowledge-based category, for example, as described in [9, 10, 11]. In their research, they used a case-based recommender to generate recommendations. This method proved its capability by generating a bundled region as its recommendation for the user. Implementation of a case-based recommender on a mobile device was also conducted by [12]. They used a third-party travel planning system to retrieve a predetermined case and combine it with the user’s constraint to generate recommendations. A different approach was taken by [13] to generate recommendations. They utilized a location-based social media’s user activity, built clusters of activity, and used it to generate recommendations. Research [14] categorized tour destinations as Tour Building Block (TBB). Before recommending a destination, the similarity between the user’s constraint and TBBs is calculated using a semantic matching algorithm. Alteration of basic case-based recommender was conducted by [15]. They extended the basic case-based recommender into a case-based conversational recommender, enabling users to generate similar cases or ask users to create their constraint. This alteration is actually a variation in the form of collecting user cases, or in a broader term known as user profile or preferences. There are other methods to collect such information, divided as implicit and explicit methods [16]. The implicit method works seamlessly when a user scrolls and views a tourist destination description that attracts him. This method keeps track of
user behavior while exploring the application. Sometimes the user is just curious about some places. This curiosity will still be taken into account regardless of whether the user is really interested or just randomly clicks something on the screen. On the other hand, the explicit method asks the user what they prefer and explicitly determines the user profile. A fixed profile such as gender, date of birth (to collect information about age), and user basic preferences need to be input just one time only at the start of the application to reduce the repetitive actions. Some dynamic preferences should be entered dynamically. Research about increasing case-based reasoning’s efficiency was conducted by [17]. Their research aims to increase efficiency in retrieving recommendation by mimicking the bee’s instinct when gathering food. By using this method, they can generate a recommendation after 500 iterations. Another efficiency effort was also conducted by [18]. They proposed a new approach to calculate similarity value between attributes and manage to outscore Support Vector Machine, Naïve Bayes, Artificial Neural Network, and Radial Basis Network by 96.69%.

This research explores and implements other approaches in case-based recommendation, where the recommendations are based on the specific user’s preferences and constraints. The user preferences and constraints are driven by the derivation of local knowledge in tourism, which is then formed into attributes of tourism categories, activities, and types of visitors.

3. Methodology

In this study, we generate recommendations by using a case-based reasoning approach. It is a subset of a knowledge-based recommender system. This method was chosen due to its capability in solving cold start problems (a problem when the system cannot draw any inferences for users or items because there is no sufficient information). Users will need to specify their query by answering some questions in the application. A query or user’s preference is then considered as a new case. Case-based reasoning solves a new case by retrieving items (in this case, the item is a tourist attraction) and adapting to fit the case [18].

3.1. Dataset

The tourist attractions which are used in this study were collected by using Google Maps API. Tourist attractions in the city of Yogyakarta, Indonesia, are used as a case study. The collected tourist attractions have three types of attributes: category, activity, and visitor type, affecting the recommendation. These attributes are derived from the specific knowledge base taken from local tourism authorities, which are then used as the base of user preferences and constraints. The values for each attribute are shown in Table 1. We have listed 217 tourist attractions and assigned values for each attribute. In our system, some attributes are likely to contain more than one value. For example, while in the park, the user can do the following activities: jogging, playing, walking around, enjoying the beauty, and relaxing. The labeling process is done by three persons who possess sufficient knowledge about the characteristics of tourist attractions in Yogyakarta.

| Attribute   | Values                                                                 |
|-------------|------------------------------------------------------------------------|
| Category    | reservoir, hill, lake, zoo, museum, garden, desert, bath, tomb, water boom, waterfall, valley, beach, park, mountain, cave, river, historical place, monument, building & structure, art gallery, temple, industrial heritage, creative art & workshop, tour park, amusement park, science, aquaria, library, culinary, mall, market, tourist village |
| Activity    | climb, swim, go cave, jogging, camping, gardening, animal hunting, fishing, boat ride, playing, go around, outdoor sport, recreation, trekking, learn art, enjoy the beauty, study history, soak, take the picture, eat and drink, relax, learn to code |
| Visitor type| solo-travel, couple, friends, family-with-kids, family-with-elder-people |
3.2. Recommender System

Figure 1 denotes the workflow of the recommender system used in this study. Starting from the “registration phase” and then followed by the “get recommendation phase”.

![Figure 1. System’s workflow.](image)

In the registration phase, users are asked to fill their profile and pick several samples of tourist attractions that they like. This process is only done once at the beginning of registration, but users can change their choices through the edit profile menu provided in the application. The sample of tourist attractions that are presented to the users is from the category attribute. The chosen items will be stored in the database and attached to the user's profile.

Each time a user decides to create a new trip plan, they are asked to specify their trip's plan, including the length of the trip, trip companion, and activities plan. Trip duration information is important to determine the number of items to be recommended. The trip companion information will be matched with the visitor type attribute, while the activities plan information will be matched with the activity attribute attached to each tourist attraction item. To make it easier, users just need to pick the answer from the application's options. These steps are necessary to gain a more precise user's specifications. The user's specification is then used to define the new case, which will be solved using the case-based recommender system.

Upon completing the user's specification, the system will retrieve similar tourist attractions items that have been stored in the database. To generate a more precise recommendation, the system needs to
calculate the similarity between tourist attraction items and user’s specifications (user’s query or new case). Equation (1) denotes the formula to calculate the similarity value. The attributes in the user’s query and tourist attraction item use a categorical value type. Therefore a different approach is needed to calculate the similarity. We use the Simple Matching Coefficient to calculate these categorical attributes in our system, as shown in Equation (2).

\[
f(t, x) = \frac{\sum_{i \in S} w_i \times \text{Sim}(t_i, x_i)}{\sum_{i \in S} w_i}
\]

where
- \( t \) = user’s query
- \( x \) = tourist attraction item
- \( i \) = index of attribute
- \( S \) = set of tourist attraction’s attribute
- \( w_i \) = weight of attribute \( i \).

\[
\text{Sim}(t_i, x_i) = \frac{M_{11} + M_{10}}{M_{00} + M_{10} + M_{01} + M_{11}}
\]

where
- \( M_{11} \) = the total number of attributes where \( t_i \) and \( x_i \) both have a value of 1
- \( M_{01} \) = the total number of attributes where the attribute of \( t_i \) is 0 and the attribute of \( x_i \) is 1
- \( M_{10} \) = the total number of attributes where the attribute of \( t_i \) is 1 and the attribute of \( x_i \) is 0
- \( M_{00} \) = the total number of attributes where \( t_i \) and \( x_i \) both have a value of 0

To give a better perspective of our proposed similarity calculation, we will demonstrate a simple similarity calculation between a user query and a tourist attraction. Consider a tourist attraction in Table 2 and a user query in Table 3. Let us assume that there are 6 kinds of activity, 4 kinds of visitor type, and 7 kinds of category. The calculation of similarity for each attribute and similarity between the user’s query and tourist attraction is shown in Table 4. We employ the Simple Matching Coefficient to count the similarity of activity attribute and category attribute. For the visitor type attribute, the similarity score is 1 if the value in the user’s query can be found in the tourist attraction or 0 if it can be found. After all the similarity calculation of all attributes is done, we finally can calculate the similarity between the tourist attraction and the user’s query.

| Table 2. Tourist attraction sample. |
|-----------------------------------|
| Attraction | Activity (w = 3) | Visitor type (w = 5) | Category (w = 4) |
|-----------|-----------------|---------------------|-----------------|
| Malioboro | Shopping, eat and drink, taking pictures | Family-with-kids, friends, couple | Monument |

| Table 3. User’s query sample. |
|------------------------------|
| Query | Activity | Visitor type | Category |
|------|----------|--------------|----------|
| Q1   | Shopping, adventure, taking pictures | Family-with-kids | Monument, science, zoo |

| Table 4. Similarity calculation. |
|--------------------------------|
| Activity | Visitor type | Category | Similarity |
|---------|--------------|----------|------------|
| 2+6-4   | 1            | 1+7-3    | (3x0.67)+(5x1)+(4x0.71) |
| (6-4)+1+1+2 | (7-3)+0+2+1 | 3+5+4   | 0.82      |
After calculating the similarity, the system will rank the possible candidate of tourist attractions based on their similarity value. Only tourist attractions with a similarity value above 75% will be shown in the application as the recommendation item. However, not all these recommended items will be presented to the user because the amount of presented items depends on the user’s planned trip duration. This design is chosen to avoid a vast amount of recommendations that may overwhelm users.

3.3. Evaluation

To evaluate our recommendation model, 10 relevant users will be involved. These 10 users involved are the users who frequently travel so that they are considered enough as representation. Each user will be asked to use the application starting from the registration to getting the recommendation with the trip duration of two days. Users are asked to create their-own similarity value on each tourist attraction that is recommended by the application. This user-created similarity value will be subtracted with the generated similarity value and called the error. The evaluation method that we choose in this study is Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE).

### Table 5. MAPE interpretation.

| MAPE Value     | Interpretation            |
|----------------|---------------------------|
| Less than 10%  | Highly Accurate Prediction|
| 11% to 20%     | Good Prediction           |
| 21% to 50%     | Reasonable Prediction     |
| More than 51%  | Inaccurate Prediction     |

3.3.1. Mean Absolute Error (MAE). The MAE calculation involves the absolute value of the errors and then dividing the total error by n [19]. MAE indicates how much our model missed in predicting the similarity value. The calculation of MAE is shown in equation 3. To conclude the overall performance of our model, we calculate the average MAE of 10 users.

\[
MAE = \frac{1}{n} \sum_{i=1}^{n} |y_i - y'_i|
\]  

where

- \(y_i\) = user’s similarity i-th value
- \(y'_i\) = generated similarity i-th value
- \(n\) = total amount of data

3.3.2. Mean Absolute Percentage Error (MAPE). Another useful evaluation is using the Mean Absolute Percentage Error (MAPE). MAPE is the mean or average of the absolute percentage errors of a prediction. Since MAPE is in percentage, it is more understandable than other evaluation metrics. Each recommended destination’s absolute error is divided by the actual similarity value provided by the user to form it into an absolute percentage error. Absolute percentage errors are summed and averaged to compute MAPE. Equation 4 provides mathematical notation to calculate MAPE. This measure is easy to interpret because it provides the error in terms of percentages. Upon interpreting the MAPE, we use MAPE interpretation by [20], which is shown in Table 5.

\[
MAPE = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{y_i - y'_i}{y_i} \right|
\]  

where

- \(y_i\) = user’s similarity i-th value
- \(y'_i\) = generated similarity i-th value
- \(n\) = total amount of data
4. Results and Discussion

4.1. Application Implementation

In this study, we implemented the recommender system on an Android application. Java programming language is used to build a mobile application. A client-server architecture is used in the recommendation process. We built a separate web service that receives the input supplied by the Android application through API. The result of the recommendation from the web service is sent to the application through API and then the Android application presents the recommendation to the user.

Figure 2 and Figure 3 display the Android implementation based on the design made in section 3.2. Figure 2(a) displays the first activity that appears when a new user uses the application. It asks the user to use their Google account to log in to the application. The application will store the email and name from the Google account. When a new user logs in to the application, they will be asked to complete their personal information (shown in Figure 2(b) and Figure 2(c)). These data will be stored along with the user’s email and name from the previous activity. Figure 2(d) represents the user preference activities, which is one of the most important keys in this study. It contains the list categories of tourist attractions (along with the sample of image) that the user needs to choose to satisfy the case-based reasoning requirement. The chosen categories will be stored in the database and will be used to help generate recommendations.

![Figure 2. Android implementation (registration phase)](image)

After completing the registration steps, users will be redirected to the home activity. This activity contains bottom navigation to navigate between home, trip's history, and profile page. The trip history contains the trips that a user has done before, and the profile page contains the user's personal information and preferences. On the home page, users can choose to create a new trip plan. Figure 3(a) shows the activity that appears when a user chooses to create a new trip. In this activity, users are asked to complete some necessary information for their trip, including its length. After that, users are asked to choose the travel companion and activities plan (shown in Figure 3(b) and Figure 3(c)). This information (also called the user's query) is necessary to generate recommendation since it will be used to calculate similarities between visitor type attribute and activity attribute on the potential tourist attraction recommendation.
After completing all the required information, the application will send a POST request containing the trip specification to the web service. The web service will also retrieve the previously stored user profile that contains information about the tourist attractions categories that suit user preference. The web service then calculates similarities between potential tourist attraction items and the user’s query. After completing the similarities calculation, the web service will send the result to the application through API. The application then presents recommendations to the user (shown in Figure 3(d)). The recommendation consists of the picture of the tourist attractions, its name, and the similarity score in percent. A higher similarity score indicates that the item suits more to the user specification.

### Table 6. User assessment results.

| User    | MAE | MAPE |
|---------|-----|------|
| Person A | 10.4 | 16%   |
| Person B | 1.79 | 2%    |
| Person C | 5.36 | 7%    |
| Person D | 2.31 | 3%    |
| Person E | 4.53 | 7%    |
| Person F | 4.8  | 6%    |
| Person G | 3.4  | 4%    |
| Person H | 3.27 | 4%    |
| Person I | 2    | 2%    |
| Person J | 3.2  | 4%    |
| Average  | 4.1  | 5%    |

### 4.2. Recommendation Result

In our experiment, we managed to gather 10 travelers to assess our recommendation model. We asked them to compare our recommendation and their-own assessment regarding the recommended tourist
attractions. We used Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE) to measure our recommendation model. The results of the test are shown in Table 6.

The MAE indicates how much error does our model generates in predicting the similarity between tourist attractions and the user’s specifications. Table 2 shows various MAE and MAPE for 10 users, with person A has the highest MAE and MAPE, which indicates that our model does not suit well to her. Despite the large MAE and MAPE on person A, the average MAE for 10 persons is only 4.1, and the average MAPE for 10 persons is 5%. With MAPE less than 10%, according to [20], this recommendation is considered highly accurate.

5. Concluding Remarks
In this study, we implemented case-based reasoning as our recommendation model on the Android application. Each tourist attraction has 3 attributes (i.e., category, activity, and visitor type), which we use to calculate the similarity with the user's query. The application utilizes a separate web service to efficiently calculate the similarity between items (tourist attractions) and the user's preference. We evaluate our recommendation model using MAE and MAPE. It is shown that from 10 users, the average MAE is 4.1 and the average MAPE is 5%. With MAPE less than 10%, this recommendation is considered highly accurate.

References
[1] Tribe J 2009 *Philosophical Issues in Tourism* ed J Tribe (Bristol: Channel View)
[2] Aggarwal C C 2016 *Recommender systems* vol I (Switzerland: Springer International Publishing)
[3] Brandt T, Bendler J and Neumann D 2017 Social media analytics and value creation in urban smart tourism ecosystems *Inf. Manag.* **54**(6) pp 703–13 DOI: 10.1016/j.im.2017.01.004.
[4] Quijano-Sánchez L, Cantador I, Cortés-Cediel M E and Gil O 2020 Recommender systems for smart cities *Information Systems* **92** 101545 DOI: 10.1016/j.is.2020.101545
[5] Menk A, Sebastia L and Ferreira R 2017 CURUMIM: A Serendipitous Recommender System for Tourism Based on Human Curiosity *Proc. of the IEEE 29th Int. Conf. on Tools with Artificial Intelligence (ICTAI)* pp 788–95 DOI: 10.1109/ICTAI.2017.00124.
[6] Khallouki H, Abatal A and Bahaj M 2018 An ontology-based context awareness for smart tourism recommendation system *Proc. of the Int. Conf. on learning and optimization algorithms: theory and applications* pp 43:1–43:5 DOI: 10.1145/3230905.3230935
[7] Herzig D, Laß C and Wörndl W 2018 Tourrec: a tourist trip recommender system for individuals and groups *Proc. of the 12th ACM Conf. on Recommender Systems* pp 496–7
[8] Amoretti M, Belli L and Zanichelli F 2017 Utravel: smart mobility with a novel user profiling and recommendation approach *Pervasive and Mobile Computing* **38** Part 2 pp 474–89 DOI: 10.1016/j.pmcj.2016.08.008
[9] Herzig D and Wörndl W 2014 A travel recommender system for combining multiple travel regions to a composite trip *CBRecSys@RecSys* **1245** pp 42–8
[10] Alptekin G I and Büyükózkcan G 2011 An integrated case-based reasoning and MCDM system for web based tourism destination planning *Expert Systems with Applications* **38** pp 2125–32 DOI: doi:10.1016/j.eswa.2010.07.153
[11] Bouhanna A, Fekih A, Abed M and Chabchoub H 2013 An integrated case-based reasoning approach for personalized itinerary search in multimodal transportation systems *Transportation Research Part C: Emerging Technologies* **31** pp 30–50 DOI: 10.1016/j.trc.2013.02.014
[12] Nguyen Q N, Cavada D and Ricci F 2003 Trip@vice mobile extension of a casebased travel recommender system *Proc. of the 2nd Int. Conf. on Mobile Business*
[13] McKenzie G and Janowicz K 2014 Activities in a new city: itinerary recommendation based on user similarity *Spatial Knowledge and Information (SKI) Canada*
[14] Ten Hagen K, Kramer R, Hermkes M, Schumann B and Mueller P 2005 Semantic matching and heuristic search for a dynamic tour guide *ENTER* pp 149–59
[15] Mahmood T, Ricci F, Venturini A and Höpken W 2008 Adaptive recommender systems for travel planning ENTER 8 pp 1–11

[16] Hanani U, Shapira B and Shoval P 2001 Information filtering: Overview of issues, research and systems User Modeling and User-Adapted Interaction 11 pp 203–59

[17] Lorenzi F, dos Santos D S and Bazzan A L 2005 Case-based recommender system inspired by social insects Retrieved on November 2020 from https://www.inf.ufrgs.br/maslab/pergamus/pubs/eniabeeCR.pdf

[18] Rezvan M T, Hamadani A Z and Shalbafzadeh A 2013 Case-based reasoning for classification in the mixed data sets employing the compound distance methods Eng. Applications of Artificial Intelligence 26 pp 2001–9

[19] Willmott C J and Matsuura K 2005 Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance Climate research 30 pp 79–82

[20] Lewis Colin D 1982 Industrial and business forecasting methods (Massachusetts: Butterworth & Co Publishers Ltd)