Intention Travel Recommendation System: A Travel Recommendation System based on Online Comments

May 22, 2024

Ching Wang[‡1] Yi-Shan, Chu[‡2]

109304015@nccu.edu.tw 111701037@g.nccu.edu.tw

1. Department of Statistics, National Chengchi University, Taiwan(R.O.C)
2. Department of Mathematical Science, National Chengchi University, Taiwan(R.O.C)

Abstract

This research presents an on-time travel recommendation system utilizing data from the Google Maps API. Facilitating the content-based filtering method based on the intention network. The recommendation algorithm incorporates linear regression for weight calculation, dynamically adjusting weights based on user preferences. Through the experiment, the researchers are able to evaluate the performance of the intention network in the recommendation system. Altogether, this work highlights the potential of integrating keyword-based intention networks in travel recommendation systems representing partial public trend on the internet.

Keywords: Recommendation system, NLP, intention network

[3]The authors contributed equally to this work and are listed in alphabetical order.

1. Introduction

Recommendation system is an information filtering methods that provides customized output from a given dataset by weighing variables based on users' interests, preferences, and choices. The recommendation system was first published by Elaine Rich[16] in 1979 to solve the problem of offering personalized book recommendations to specific types of readers based on their answers to questions. Various recommendation systems have been developed sequentially since the 1990s and have been introduced in many applications, such as web browsers and e-commerce platforms .[8]

The emergence of new technology has significantly impacted traditional travel agencies, which previously relied on personal recommendations, forcing them to adapt. Online travel agencies, such as Agoda and Trivago, now use recommendation systems to provide personalized tourist spot recommendations to users. Compared to traditional travel agents, online travel agents have a better reputation and services due to their 24/7 availability and personalized design.[6]

In recent years, Natural Language Processing (NLP), encompassing large artificial intelligence models trained to understand and process natural language, has seen rapid development. Beyond their original usage, this research employs NLP techniques to construct intention networks to provide customized knowledge for recommendation models and improve user experiences. By obtaining data from Google Maps, an open-to-public rating and review platform, the research aims to provide fair and detailed tourist information to the recommendation system, offering users an equitable, comprehensive, and public-based system for selecting tourist spots.[20]

2. Literature Review

a. Recommendation System

Recommendation system helps users find customized preferred choices based on different filtering methods, and it can basically be built using two types of approaches: content-based filtering method and collaborative filtering method.

1. Content based filtering method
   The content-based filtering method is only based on the product information, which provides similar content to the users based on their choices. The similarity level is designed by different algorithms. According to the design of the similarity algorithm, the designer can identify relationships between products and perform classification and recommendation [12]. The content-based filtering method is often employed in industries with strict customer privacy policies, such as e-commerce platforms. By utilizing this method, recommendations can be made using only product information and customers' past data. However, a disadvantage lies in the potential inequality of customer ratings, which cannot guarantee accuracy and fairness[8].

2. Collaborative filtering method
   The collaborative filtering method is based on users' behavior and heavily relies on user datasets to generate recommendations. It calculates the similarity between a new user and

the original user group, and recommends products that users with the highest similarity have previously liked [7]. This method is often applied in e-resources and search engines, where users' browsing histories and preferences can be easily obtained. However, the cold start problem arises since the system requires enough data input to find interactions with the original user group.[8]

b. Recommendation System for Travel

Recommendation is widely used in digital applications, including the travel industry. Logesh Ravi et al., in 2016, evaluated social network-based recommendation systems while considering different filtering methods, functionalities, interface, and artificial techniques. [15] Majid, A. et al., added the geotagged social media data mining into the research, which can help predict foreign tourist spots for users based on past data. [9] Through the rapid development of NLP techniques, text mining and LLM are gradually being combined with travel recommendation system design. Asaithambi et al. provided a design for a thematic travel recommendation system by performing text mining and data mining on destination information and building various Spark models to evaluate the performance. [1] In this research, we incorporated keyword extraction into the travel recommendation system to provide a fair and public user experience.

c. Keyword Extraction

In the field of NLP(Nature Language Processing), keyword extraction is used to identify and extract most relevant and significant words or phrases from a text. These keywords are intended to capture the main topics of an article, which is particularly useful when reading a large number of articles, such as restaurant reviews. Keyword extraction methods can be roughly put into two main categories:

1. Statistical approaches
   Statistical approaches rely on the frequency and distribution of terms within the documents. These methods are often used due to the computationally efficient. Examples of such methods include TF-IDF(Term Frequency-Inverse Document Frequency)[18], TextRank[11] and RAKE(Rapid Automatic Keyword Extraction)[17].

2. Linguistic and machine learning approaches
   Linguistic and machine learning approaches utilize advanced technologies in computational linguistics and artificial intelligence to identify the keyword of a document. These methods often outperform simple statistical methods by incorporating syntactic, semantic, and contextual information. The linguistic methods such as POS(Part-of-Speech) tagging[10] and NER(Named Entity Recognition)[19] are combined with machine learning methods like SVM(Support Vector Machine)[3], LDA(Latent Dirichlet Allocation)[2], etc. As the thrive of deep learning, the LM/LLM(Language Models/Large Language Models) are applied to these kind of tasks. Some famous pre-traine language models are used to extract the keywords, such as GPT(Generative Pre-trained Transformer)[14], BERT(Bidirectional Encoder Representations from Transformers)[4], and LLAMA (Language Model for Query

Auto-Completion)[13]. Fine-tuning the LMs is highly beneficial for tasks in specific domains. However, it would cost lots of computational resources and hence it's not easy to implement.

3. Dataset

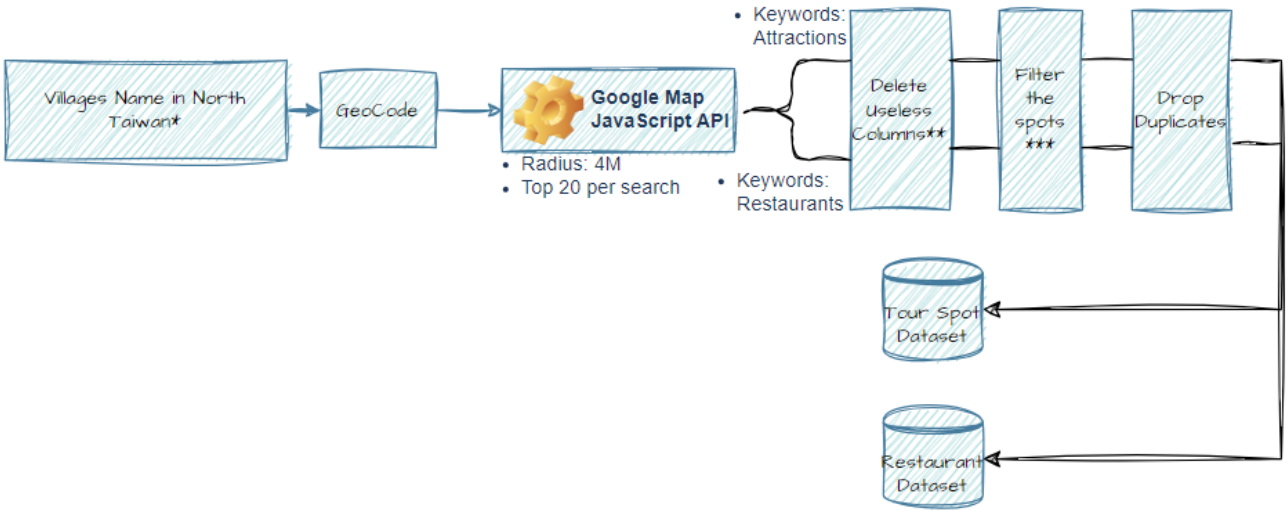

Figure. 1: The algorithm of building dataset.

* :North Taiwan includes Taipei, New Taipei City, Keelung, Yilan, and Taoyuan. ** :Only columns mentioned in Table 1 are used. *** :Spots not geographically located in the north of Taiwan are filtered out.

a. Data Resource

Aiming to build an on-time travel recommendation, this research use Google Map services, which provides Maps JavaScript API for users to collect traffic, tourism, route and attractions related data. From Google database, this research contains two dataset, one is restaurant dataset, and the other is attraction dataset. This two datasets both contain four groups of attributes, namely Business Status Attributes, Customer Review Attributes, Location Attributes, Serve Type Attributes. We use only Customer Review Attributes, Serve Type Attributes, Serve Type Attribute, `name`, `business_status`, `editorial_summary` and `opening_hours` in Business Status Attributes. The processing algorithm is providing as Figure. 1

Category	Attributes
Business Status Attributes (7)	`place_id`, `name`, `business_status`, `type`, `editorial_summary`, `formatted_phone_number`, `opening_hours`
Customer Review Attributes(4)	`rating`, `user_ratings_total`, `price_level`, `reviews_text`
Location Attributes(2)	`formatted_address`, `location(lat,lng)`
Serve Type Attributes(8)	`wheelchair_accessible_entrance`, `dine_in`, `delivery`,`serves_breakfast*`, `serves_brunch*`, `serves_lunch*`, `serves_dinner*`, `serves_vegetarian_food*`

Table. 1: Attributes of Dataset

* : variable is deleted in data processing

b. Data Processing

To create a system that provide good recommendation, we need to filter out data points with low `rating` and low `user_rating_total`. After reviewing the dataset, we decided to remove entries with `rating` below three and `user_rating_total` below 10% of the total.
Using the Kolmogorov—Smirnov Test introduced by Kolmogorov and Smirnov in 1951, we analyzed the datasets and observed a change in the distribution of the restaurant dataset but not in the tourism dataset. However, this change in distribution is not expected to impact the performance of the recommendation system.

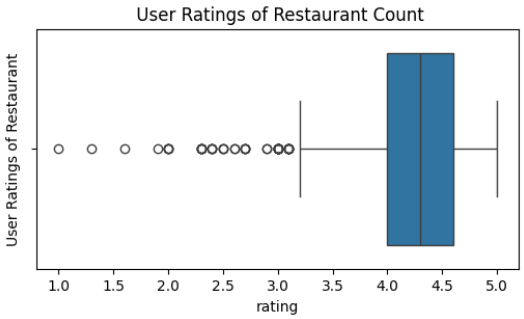

Figure. 2: Box plot of restaurant user rating

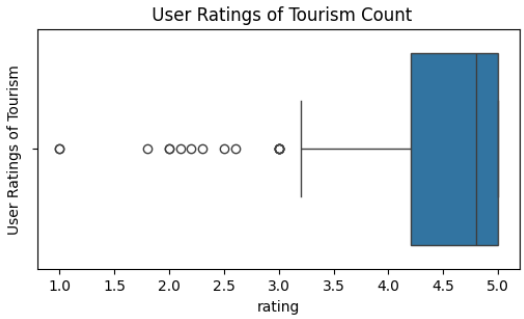

Figure. 3: Box plot of tourism user rating

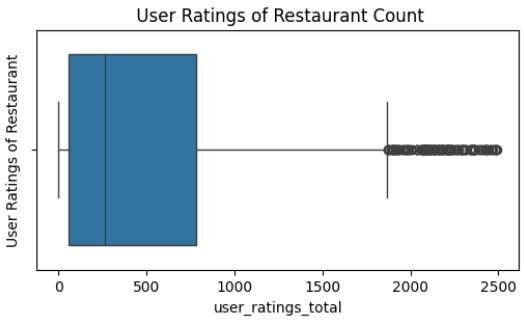

Figure. 4: Box plot of restaurant user rating total

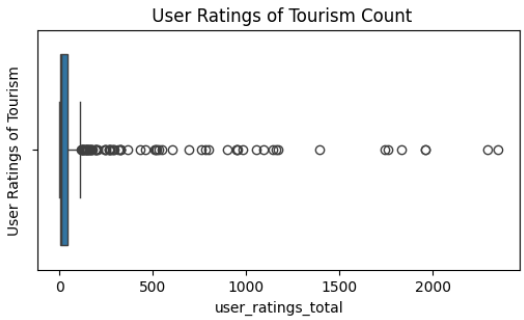

Figure. 5: Box plot of tourism user rating total

Define Attraction Operating Times

Upon counting the number of `serves_breakfast`, `serves_brunch`, `serves_lunch`, `serves_dinner`, and `serves_vegetarian_food`, we discovered that more than 67% of spots claim to serve all meals throughout the day, which seems impossible. Due to the unreliability of these four vectors, we have decided to use `opening_hours` to specify the opening times of attractions. This change aims to enhance the user experience. The specific opening hours have been categorized as `Morning`, `Evening`, `Night`, and `All Day`, with plots reflecting these classifications at 12 pm and 6 pm for added convenience. According to the `opening_hours`, the opening times are further classified based on opening weekdays, allowing us to determine the spots' opening hours on each specific workday.

4. Methodology

In this work, we build up an recommendation system based on the data of Google Map. The overview process is shown in Figure.6. At first, user has to input prefer district and weekday.(e.g. Zhongshan District, Taipei City. Friday.) Then, the system will have a first recommendation for morning base on our recommendation algorithm. It will recommend 5 tour spots and 3 restaurants for each tour spots. User has a very flexible choices for a tour spots and a restaurant spot based on the selected tour spot.

1. CASE 1: Both of the tour spots and restaurants are chosen by users. Then, the system will have a recommendation table for lunch and so on.

Tourist Spot Name	Nearby Restaurant 1	Nearby Restaurant 2	Nearby Restaurant 3
Tour A	Restaurant1	Restaurant2	Restaurant3
Tour B	Restaurant4	Restaurant5	Restaurant6
Tour C	Restaurant7	Restaurant8	Restaurant9
Tour D	Restaurant10	Restaurant11	Restaurant12
Tour E	Restaurant13	Restaurant14	Restaurant15

Table. 2: Example of Recommendation Table

2. CASE 2: Either tour spot or restaurant are chosen by users. Then, the system will have a recommendation based on the chosen spot, which will locate the neighborhood of the chosen spot. The range of the neighborhood is given by the user, such as 10 km.

3. CASE 3: Neither tour spot nor restaurant are chosen. Then the system will back to CASE 1.

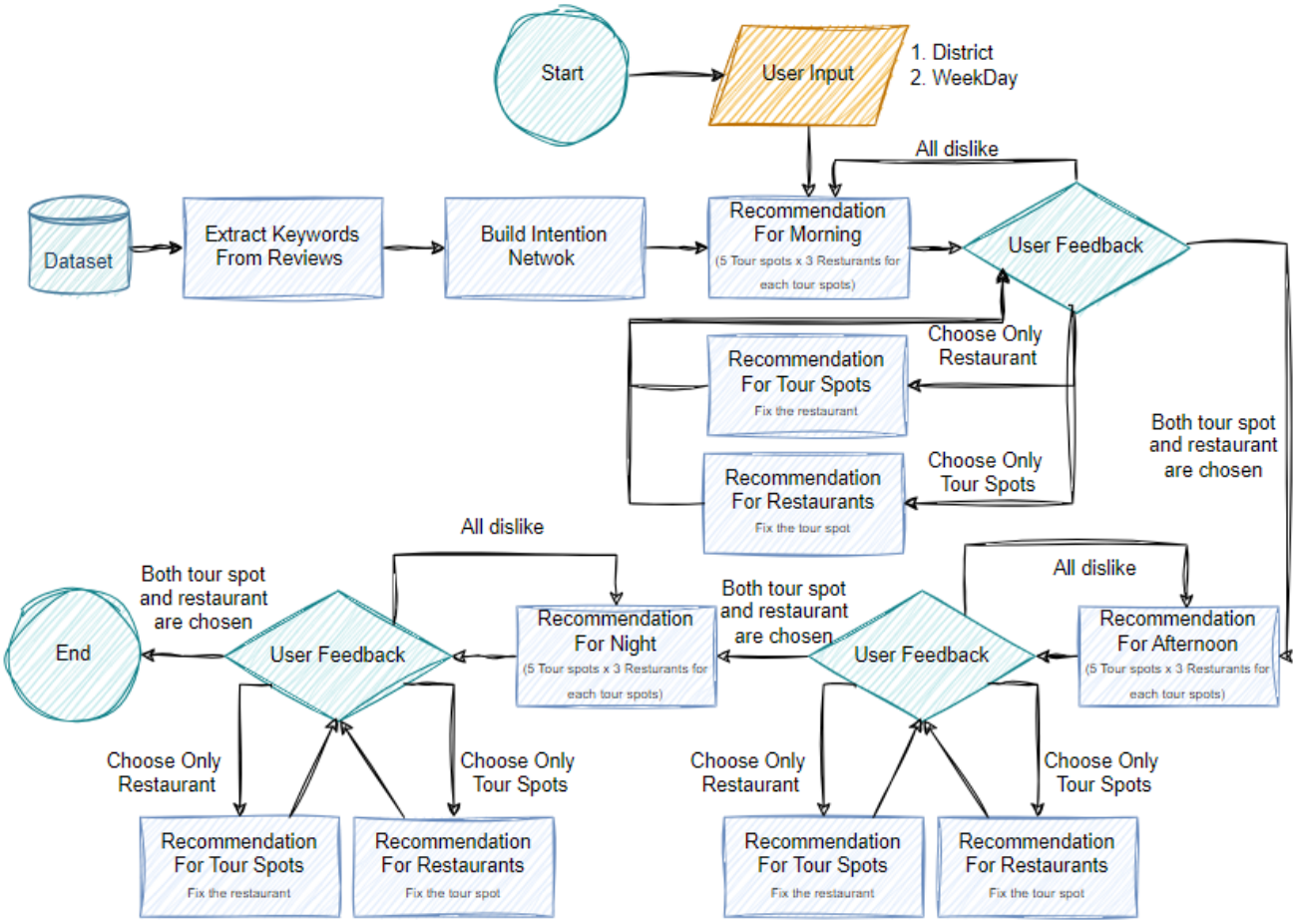

Figure. 6: The overview architecture of our recommendation system.

The process of recommendation in this work is typical. However, we published an innovative methodology to improve the performance of recommendation base on the condition that there doesn't exists any data of users. The recommendation will based on the reviews of each spots in Google Map. We build up an intention network based on the keywords extracted from the reviews to pool candidates for each recommendation.

a. Keyword Extraction

To extract keywords from the `reviews`, several keyword extraction approaches discussed previously can be used, including TF-IDF, TextRank, and KeyBERT[5]. In this work, TextRank is discussed.

TextRank

TextRank[11] is an unsupervised graph-based ranking algorithm for keyword extraction, inspired by Google's PageRank algorithm. It is particularly effective for extracting keywords and phrases from text documents. Here is an overview of the TextRank algorithm:

1. **Text Preprocessing**:
   - Tokenize the text into individual words.
   - Remove stop words (common words like "the", "and", etc. that do not carry significant meaning).
   - Stem or lemmatize the words to their base forms.
2. **Construct the Graph**:
   - Represent the processed text as a graph $G = (V, E)$, where each vertex $v \in V$ is a word.
   - Add an edge $e \in E$ between two vertices if the corresponding words co-occur within a fixed-size sliding window in the text. The weight of the edge can be the number of co-occurrences.
3. **Score the Vertices Using PageRank**:
   - Initialize each vertex with an equal score.
   - Update the score of each vertex using the formula:

$$S(v) = (1 - d) + d \sum_{u \in \mathtt{adj}(v)} \frac{S(u)}{|\mathtt{adj}(u)|}$$

     where $S(v)$ is the score of vertex v, d is a damping factor (usually set to 0.85), and $\mathtt{adj}(v)$ is the set of vertices adjacent to v.
   - Iterate the scoring process until convergence (scores do not change significantly between iterations).
4. **Extract Keywords**:
   - Rank the vertices based on their final scores.
   - Select the top N vertices as the most significant keywords.

TextRank effectively captures the importance of words based on their relationships with other words in the text, making it a robust method for keyword extraction.

b. Intention Network

Due to the lack of user's data, it's not possible to do the collaborative filtering method. However, we've come up with an alternative graph-based method in this work, the Intention Network. First, extract keywords from the reviews of each spots. Then take the keywords as the source node of each graph, and if the spot have the keywords, there is an edge(relationship) connect from the keywords to the spot. Take Figure.7 for example, all of the spots A,B,C,D,E have the keyword, so they are connected in a graph. Note that each spots can emerge in at most 10 graphs because we only extract 10 keywords per spot in this work. After building up all networks base on the Google reviews, we pool the spot candidates based on this intention network. Before every recommendation, we will select all spots that are connected to the previous choice of user. Then, we pool all potential spots with connections, where duplicates are permitted, and the spots with the highest weight are displayed.

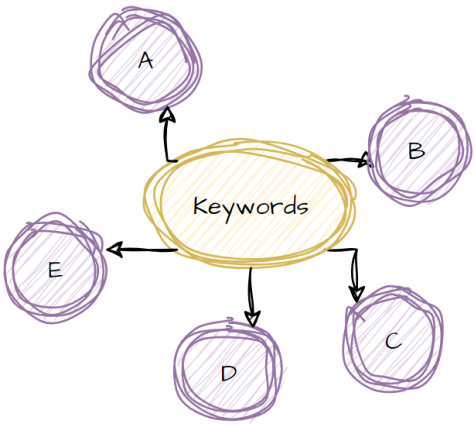

Figure. 7: Examples of Intention Network based on keywords, where A,B,C,D,E are the spots that have the keywords.

c. Recommendation Algorithm

Given that the dataset obtained from the Google Maps API encompasses solely product-based data, such as information pertaining to tourism and restaurants, and lacks user-specific information, this research employs a content-based recommendation approach for designing the recommendation system. The recommendation system can be delineated into two components: the content-based recommendation module and the weighted algorithm module.

1. Content-based recommendation
   In this research, the recommendation is based on data information of tourist spots and restaurants. Subsequent to the selection of the initial spot, the system start calculating the similarity of each spot. However, in order to facilitate the updating of similarities and account for the bonus associated with continuously selected features at the same time, this research apply linear regression to generate weighted scores, instead of similarity score. The

ranking of spots is determined by the assigned weight, which is presented to the customer.

$$\begin{aligned}\texttt{weight}_i = w_{\texttt{rating}} \cdot \texttt{rating\_norm}_i + w_{\texttt{ratings\_total}} \cdot \texttt{user\_ratings\_total\_norm}_i \\ + w_{\texttt{district}} \cdot \texttt{district\_norm}_i + \epsilon(0,1)\end{aligned}$$

2. Weighted algorithm
   The weighted is subject to updates based on user preferences and a random variable that introduces an element of surprise, allowing lower-ranked spots to potentially rise in the rankings. For first-time customers, the `district` weight is assigned a value of 1, enabling users to specify a preferred area to visit. However, for subsequent choices, the `district` weight is set to 0. After the initial choice, the $w_{\texttt{district}}$ is abandoned, and the system transitions to the $w_{\texttt{distance}}$ weight. The $w_{\texttt{district}}$ is calculated in euclidean distance, and only spots which `district` smaller than the users given distance are under consideration. When a user selects a specific spot, other spots sharing the same `rating`, `ratings_total`, `distance`, or `keywords` receive a weighted bonus. Notably, if a variable is selected multiple times, say n times, the bonus weight is multiplied by n.
   The spot weighted calculation is:

$$\begin{aligned} w_{\texttt{a*}}[i+1] &= w_{\texttt{a}}[i] \cdot (1 + 0.3 \cdot (n+1)) \\ w_{\texttt{b**}}[i+1] &= w_{\texttt{b}}[i] \cdot 1.1 \\ n &= \text{selected count} \end{aligned}$$

*a is `rating`, `ratings_total`, `distance`, and `keywords`, for the spot that was chosen.

**b is `rating`, `ratings_total`, `distance`, and `keywords`, for the spot that was not chosen.

5. Results and Analyzes

Experiments

In this work, we have a simple example of the recommendation system. We use TextRank algorithm to extract top 10 keywords for each spots and build up the intention network to process the recommendation. Based on the dataset we collected our own, we build up the intention recommendation system base on the design of Figure.6. Also, to have a comparison, we also build up a single recommendation system without the intention network but utilized the same recommendation algorithm. A small test in a group of 25 people is to be conducted. It's expected that the intention recommendation system have a more relevant and better recommendation than the single recommendation system without the informant of intention network.

6. Conclusion

Future Works

(a) Different keyword extraction method can be used to improve the performance of this method.

(b) The lack of user data poses a challenge for collaborative filtering methods. If the user data are given, it's believed that it can have a further improvement of the performance.

(c) Due to the constraint of the Google Map API, we can only have the latest five comments of each spots, which may lead to bias. Also, the number of instances is insufficient to support a complete recommendation process. If we can collect more data, the performance would be better.

(d) In this work, only the text in reviews is used. But the name and editorial summary of each spots may also provide useful information. Hence, in future work, we can also extract keywords in the name and summary of each spots. In addition, the picture of the menu will also give more information. Here, the OCR(Optical Character Recognition) method can be used to extract the words in the picture of the menu.

(e) The number of experiments is not enough to support the statement that adding intention will give better performance. Hence, we are still having more experiments on different people(different in age, gender, racial, country, social background, etc).

(f) In the recommendation algorithm, a lot of parameters are pre-defined. The changes of these parameters will have a different recommendation. Therefore, finding best parameters will also improve the performance of the system.

7. References

[1] S. P. R. Asaithambi, R. Venkatraman, and S. Venkatraman. "A thematic travel recommendation system using an augmented big data analytical model". In: *Technologies* 11.1 (2023), p. 28.

[2] David M Blei, Andrew Y Ng, and Michael I Jordan. "Latent Dirichlet Allocation". In: *Journal of Machine Learning Research* 3 (2003), pp. 993–1022.

[3] Corinna Cortes and Vladimir N Vapnik. "Support-vector networks". In: *Machine learning* 20.3 (1995), pp. 273–297.

[4] Jacob Devlin et al. "BERT: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[5] Maarten Grootendorst. *KeyBERT: Minimal keyword extraction with BERT*. `https://github.com/MaartenGr/KeyBERT`. 2020.

[6] R. Hassannia et al. "Web-based recommendation system for smart tourism: Multiagent technology". In: *Sustainability* 11.2 (2019), p. 323.

[7] Y. Koren, S. Rendle, and R. Bell. "Advances in collaborative filtering". In: *Recommender systems handbook*. Springer, 2021, pp. 91–142.

[8] Jie Lu et al. "Recommender system application developments: a survey". In: *Decision Support Systems* 74 (2015), pp. 12–32.

[9] A. Majid et al. "A context-aware personalized travel recommendation system based on geotagged social media data mining". In: *International Journal of Geographical Information Science* 27.4 (2013), pp. 662–684.

[10] Mitchell P Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. "Building a large annotated corpus of English: The Penn Treebank". In: *Proceedings of the workshop on Human Language Technology*. Association for Computational Linguistics. 1993, pp. 313–318.

[11] Rada Mihalcea and Paul Tarau. "TextRank: Bringing order into text". In: *Proceedings of the 2004 conference on empirical methods in natural language processing*. 2004, pp. 404–411.

[12] P. Nagarnaik and A. Thomas. "Survey on recommendation system methods". In: *2015 2nd international conference on electronics and communication systems (ICECS)*. IEEE. 2015, pp. 1603–1608.

[13] Amrith Parmar et al. "LLAMA: A large-scale, open-domain language model for query auto-completion". In: *arXiv preprint arXiv:2103.06561* (2021).

[14] Alec Radford et al. "Improving language understanding by generative pre-training". In: *OpenAI Technical Report* (2018).

[15] L. Ravi and S. Vairavasundaram. "A collaborative location based travel recommendation system through enhanced rating prediction for the group of users". In: *Computational intelligence and neuroscience* (2016).

[16] E. Rich. "User modeling via stereotypes". In: *Cognitive science* 3.4 (1979), pp. 329–354.

[17] Stuart Rose et al. "Automatic keyword extraction from individual documents". In: *Text Mining: Applications and Theory*. Wiley Online Library, 2010, pp. 1–20.

[18] Gerard Salton and Christopher Buckley. "Term-weighting approaches in automatic text retrieval". In: *Information processing & management* 24.5 (1988), pp. 513–523.

[19] Erik F Tjong Kim Sang and Fien De Meulder. "Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition". In: *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*. 2003, pp. 142–147.

[20] W. X. Zhao et al. "A survey of large language models". In: *arXiv preprint arXiv:2303.18223* (2023).