User Adaptive Intelligent System

LeshanWang 2019110023

1. Introduction

1.1 Sustainability Aspect

The user adaptive intelligent system of this project aims to provide personalized food and restaurant suggestions to users to help them achieve a sustainable lifestyle. Food, as a daily subject, closely relates to one's body health. According to WHO, a healthy diet helps form protection against malnutrition, noncommunicable diseases, and other conditions (WHO, 2022). In the bigger picture, food consumption can have a significant impact on the global environment. For example, massive meat consumption has become a major concern for its impact of livestock on global warming and environmental degradation (Marenzi et al, 2022). To summarize, a sustainable diet could be both beneficial to personal gains and environmental protection.

1.2 Personalization

As food choices are rather personal subjects, a user adaptive recommendation system is necessary to provide the user with relevant and timely suggestions, which should be highly adapted to individual's personal preferences and habits. A popular definition of personalization by Anthony Jameson is that personalization is conducted of adaptability, adaptation, and anthropomorphism (Jameson. A, 2011). These three aspects aim to adapt a system based on decisions influenced by user's actions also external circumstances. As the user is able to customize aspects of the system according to their needs, the system modifies itself to suit user better, and provides human-machine friendly interactions (Sili. M et al, 2016). Therefore, a personalized system will provide the user with more appropriate, accurate, and satisfiable content; together, works better towards individual goals.

1.3 Requirement Analysis & Derivation

The requirement of this project is divided into three categories, which are functional requirements, performance requirements and system technical requirements. For each requirements, they are further derived by transferring results from analysis to logical functional architectures of the system, thus finally construct a logical model specifying every key feature and their behaviors of the system. In short, derivation of requirements is the process of making implicit requirements explicit, enabling them to be stated and captured, then to be scientifically assessed (NIST, 2016).

1.4 Requirement Description

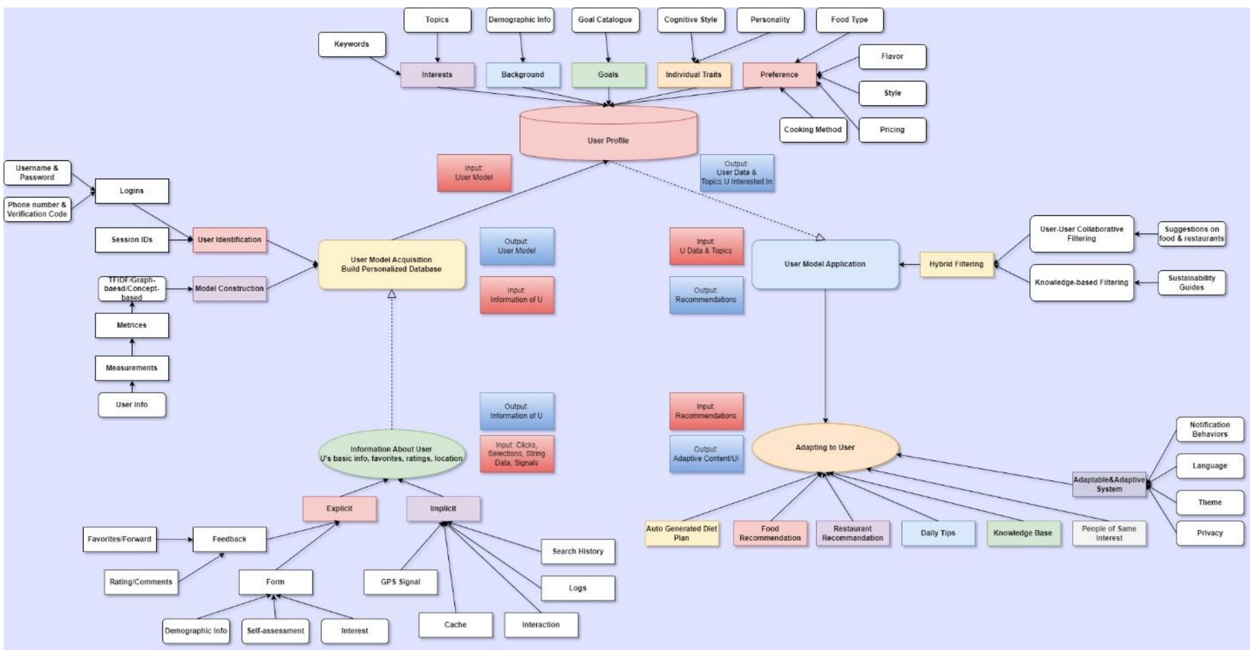
1.4.1 Functional: Timely recommendation, user-adaptable & self-adaptive, highly accessible, light/dark mode, multi-language, reminder, AI powered to do-list, calendar, map.

1.4.2 Performance: Runs smoothly on prevailing mobile devices, low power cost, low memory consumption, low CPU & GPU impact.

1.4.3 Technical: Algorithm, runs on Android/iOS, optimized garbage collect system, CPU muti-thread programming, local & cloud databases, GPS signal acquisition, Bluetooth.

2. Prototype Architecture

2.1 Diagram



2.2 User Data Collection

Input: Clicks, selections, string data, signals.

Output: Information of user.

2.2.1 Explicit: Data collected by acquiring user’s basic info (filling form when registry), including demographic, self-assessment, and interest; active feedbacks such as ratings, comments, forwards, and favorites.

2.2.2 Implicit: Data collected by analyzing user's search histories, interactions, geographical location (GPS).

2.3 User Model Representation

Input: User model.

Output: User data and topics user interested in.

User models are represented by interests, background, goal, individual traits, and preference. Combining these aspects enables the system to predict and identify topics that the user might be willing to follow accurately.

2.3.1 Interests: Keywords and topics that the user frequently searches for.

2.3.2 Background: Demographic information is a stable set of data about user's previous experience, such as their profession, name, age sex, nationality, also their opinions.

2.3.3 Goal: The most challenging aspect, uses a pre-defined goal list where user's goals are marked inside, then used for adaptation.

2.3.4 Individual Traits: Considering cognitive style and personality, together define user as an individual.

2.3.5 Preference: Similar to interest but more detailed, covers user's attitude towards different cooking methods, pricing, diet style, flavor and type of food

2.4 Modeling Method

Input: Information of user.

Output: User model.

Acquisition of user model follows two steps. Firstly, the user is identified by login or session IDs. Then based on user information extracted previously, measurements are taken and metrices are derived by combining the measurements. According to different scenarios, various methods are implemented to construct the user model.

2.4.1 TFIDF: Generates list of keywords.

2.4.2 Graph-based Filtering: Overlaying Graph models.

2.4.3 Concept-based Filtering: Generates list of concepts.

2.5 User Model Application

Input: User data and topics user interested in.

Output: Recommendations.

Implementation of hybrid filtering, using User-User Collaborative Filtering to generate suggestions of food and restaurants; using Knowledge-based Filtering to generate guidelines on sustainable lifestyle.

2.6 User Adaptive Interface

Input: Recommendations.

Output: Adaptive content and UI.

After generation of recommendations, the interface will adapt itself to promote various subjects to user, such as diet plan, food and restaurant recommendation, daily tips, knowledge base, and people of same interest. The UI is also self-adaptive to display the information, and adaptable to user as to modify menu, theme, language, privacy, and notification behavior.

3. Outline of The Recommender Method

The system uses the hybrid method of User-User Collaborative Filtering and Knowledge-based Filtering.

3.1 User-User Collaborative Filtering

Collaborative filtering is a well-known unsupervised recommender method used by many mainstream websites, such as Netflix, Amazon, and LinkedIn. According to Netflix Research, over 80% of video content people watch comes from Netflix's recommendation algorithm, which mainly uses collaborative filtering (Netflix Research, 2021). CF tackles the similarities between users to perform recommendations, give suggestions to a user based on the likes and dislikes of similar ones. In this particular case, the method recommends food and restaurants to the user based on what other similar users have liked.

The background data required by CF is the meta data of food, restaurants, meta data of similar users. The input data is the user data, including interests, demographic background, goals, individual traits, and preference. The CF utilizes user data to identify similar users from user database, and inputs data of items (food / restaurants) and users into a user-item matrix to perform the memory-based approach. CF calculates cosine similarities between each user to determine neighbors; then consider center-based neighborhood with size 4, calculates the weighted sum and finally implement association rules recommendation.

3.2 Knowledge-based Filtering

Knowledge-based filtering is an alternative solution to the cold start problem (Burke. R, 2000), which is faced by CF, as it makes recommendations based on specific queries made by the user instead of user’s rating. It is also useful when the item space is complex, for example a conceptual diet plan with professional knowledge basis.

The VITA (Virtrualis Tanacsado) financial services recommendation environment implements this filtering technique. It has been proved to be a successful and efficient application of knowledge-based recommender technologies in commercial environment with clear positive impact on existing business processes (Felferning. A et al, 2007).

The background data required by Knowledge-based filtering is the domain knowledge and meta data of food / restaurants, including food properties, cooking method, flavor, style / location, pricing, schedules, ratings, etc. The input data is the user data and topics, together are mapped into a user profile. After getting user data, the algorithm will search domain knowledge for relative rules and construct them into a single database query. The query is then used to retrieve food and restaurants from the database, then presented to the user. The user can critique the results by tightening and loosening parameters, thus, to create new queries to conduct new searches.

4. Critical Review

The proposed architecture prototype has several computational and human-factor strengths and weaknesses that need to be addressed.

4.1 Strengths

4.1.1 Computational

The hybrid method of User-User Collaborative Filtering and Knowledge-based Filtering can overcome the cold start problem faced by CF solely.

4.1.2 Human Factor

The explicit data collected from the user is reliable, complies with privacy regulations and gives the user control.

4.2 Weakness

4.2.1 Computational

The User-User Collaborative Filtering faces scalability problems when the dataset is extremely large, the effectiveness of system will be challenged.

4.2.2 Human Factor

The explicit data collected from the user can be obtrusive and might overlooks dynamic changes.

5. Conclusion

In this document, the architecture of the user adaptive intelligent system of sustainable lifestyle is discussed and assessed. The system is designed to give timely and self-adaptive content on food and restaurants with the topic of sustainability to users with interest. It implements a hybrid recommender method of User-User Collaborative Filtering and Knowledge-based Filtering, which avoids the cold start problem, however, still needs to be further optimized to minimize the impact of scalability problem.

6. Reference

6.1 WHO, 2022. 'Promoting Healthy Diet', [Online]. Available at: https://www.who.int/westernpacific/activities/promoting-healthy-diets (Accessed 6 November 2022)

6.2 Cinzia Di Novi, Anna Marenzi, 2022. 'Improving health and sustainability: Patterns of red and processed meat consumption across generations', Health Policy, [Online]. Available at: https://www.sciencedirect.com/science/article/pii/S0168851022002743 (Accessed 6 November 2022)

6.3 Anthony Jameson, 2011. 'Systems that adapt to their users', [Online]. Available at: Systems That Adapt to Their Users | v3 | Human Computer Interaction Ha (taylorfrancis.com) (Accessed 7 November 2022)

6.4 Sili, M., Garschall, M., Morandell, M., Hanke, S., Mayer, C., 2016. 'Personalization in the User Interaction Design', [Online]. Available at: https://link.springer.com/chapter/10.1007/978-3-319-39510-4_19 (Accessed 8 November 2022)

6.5 NIST, 2022. 'Systems Security Engineering', [Online]. Available at: derived requirements - Glossary | CSRC (nist.gov) (Accessed 8 November 2022)

6.6 Netflix Research, 2021. 'Figure out how to bring unique joy to each member', Recommendations, [Online]. Available at: https://research.netflix.com/research-area/recommendations (Accessed 9 November 2022)

6.7 Robin Burke, 2000. 'Integrating Knowledge-based and Collaborative-filtering Recommender Systems', [Online]. Available at: Integrating Knowledge-Based and Collaborative-Filtering Recommender Systems (aaai.org) (Accessed 10 November 2022)

6.8 A. Felfernig, K. Isak, K. Szabo, and P. Zachar, 2007. ‘The VITA financial services sales support environment’, In Proceedings of the 19th national conference on Innovative applications of artificial intelligence - Volume 2 (IAAI'07). AAAI Press, 1692–1699, [Online]. Available at: https://www.aaai.org/Papers/AAAI/2007/AAAI07-274.pdf (Accessed 12 November 2022)

7. **Prototype Presentation Link**

   https://clipchamp.com/watch/i5KZVBM7tVT