Syed Imran Ali

DEMO HARNESS

Service Curation Layer v2.5

(Recommendation Interpreter)

25 DEC 2016

Table of Contents

1. Overview
2. Recommendation Interpreter
   1. Introduction
   2. Capabilities
   3. Pre-requisite data files
      1. Physical activity data files
      2. Food item data files
   4. Structure of data files
   5. Food-item recommendation
   6. Admissible rules for Recommendation Interpreter
3. Running Scenario
4. Recommendation Interpreter Architecture
5. Overview

Recommendation Interpreter (RI) is envisioned as a contextual information processing unit of a larger recommendation generating system. In the current, concept-of-proof implementation, RI deals with two kinds of recommendations i.e. physical activity based recommendation and food-items recommendation. The main purpose of RI is to complement the recommendation generated based on user’s physiological requirements or conditions. RI takes into account contextual information of the user in terms of location of the user, user’s current activity, weather conditions, and emotional state of the user. Physical activity recommendations are based on a novel concept called “Contextual Matrix”. Contextual Matrix processes users’ preference data and in turn produces cross-contextual global patterns. This contextual matrix is populated using users’ surveyed data.

1. Recommendation Interpreter
   1. Capabilities

RI in the current implementation has following features:

1. Assess user’s interruptibility
2. Evaluate recommendation for its contextual suitability
3. Provide alternative recommendations if so required
4. Provide explanatory sentences along with audio/visual aids for the generated recommendations
5. Provide food based recommendations
   1. Pre-requisite data files

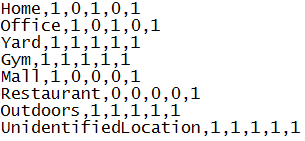
RI requires a number of data files to perform aforementioned activities. Data files are logically categorized into two types based on the nature of recommendation i.e. physical activity data files and food-item data files.

* + 1. Physical activity data files
       - Location.txt
       - HLC.txt
       - Weather.txt
       - Emotion.txt
       - Disable.txt
       - Preferences.txt
       - Explanation.txt
       - URL.txt
       - Postprocessing.txt
       - RecommendationMapper.txt
    2. Food-item data files
       - FoodCatToFooditems.txt
       - FooditemsToNutrient.txt
       - NutrientToCategory.txt
       - UserChoices.txt

Note: For the current implementation these files are required to be provided. However, based on the system requirements, this code can be extended to read/write from any other repository. Moreover, current version provides a proof of concept implementation of the RI, in subsequent versions extendibility of the component along with handling multiple recommendations will be addressed following the best practices of software design.

* 1. Structure of data files

Almost all of the files share the same *key-value* pair structure. Following figure depicts structure of Location.txt file:

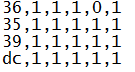


For example first string “*Office,1,0,1,0,1*” depicts a contextual scenario for location. Each bit represents a binary decision. Currently RI deals with five recommendations namely ‘*Walking’*, ‘*Running’*, ‘*Stretching’*, ‘*Cycling’*, and ‘*Sitting’*. Recommendations are treated in the same order as mentioned. For example in ‘Office context ‘*Running’* and ‘*Cycling’* can’t be performed. These data files are based on surveyed data. Users may change these values according to their own suitability and desired results. Contextual matrix will behave as per the data fed to it.

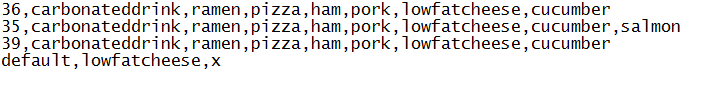
Data files are provided in:



All the pre-requisite data files are in the aforementioned package. It is important to note that in the present build (version 1.0) user’s information has to be entered manually in the *Preferences.txt* file (for physical activity) and *UserChoices.txt* (for food-item recommendation).



Where, *UserChoices.txt* includes those food-items which are disliked by the user in the given list provided in *FooditemsToNutrient.txt.*

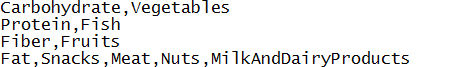


For example userid ‘*36*’ doesn’t like ‘*Cycling’* and doesn’t prefer food-items such as “*Carbonated drink, Ramen, Pizza, Ham, Pork, Low Fat Cheese, and Cucumber*”.

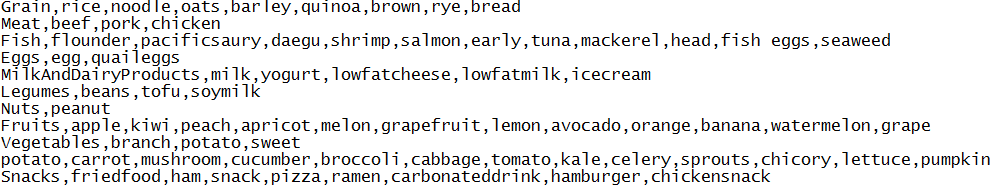
* 1. Food-item recommendation

In the current version RI cater for following nutrient categories:

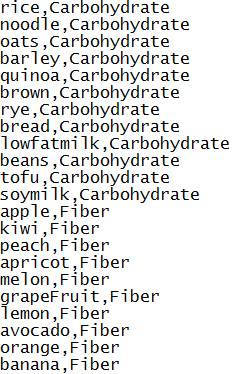
* + 1. Carbohydrate
    2. Protein
    3. Fiber
    4. Fat



Following are the mappings from food categories to food items



Following is a snippet for food-item to nutrient mappings:



Note: In the aforementioned snippets first word is treated as *key* while rest of the string is dealt as *value*, as in the key-value pair. Naming conventions must be followed as depicted in the data files.

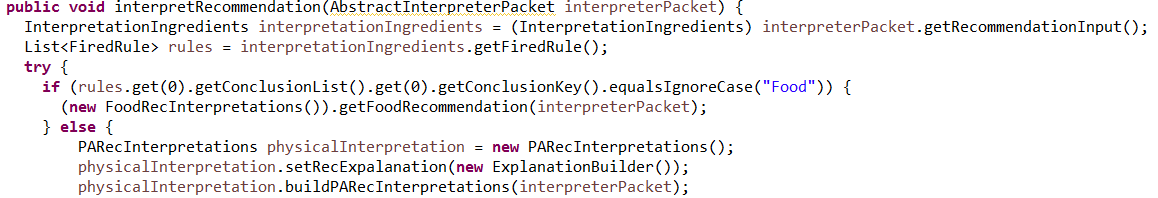
Following is a sample output for food-item recommendation:



Where, SNS trend value is followed by the recommended food-item, which may be used for visualization purposes as well.

* 1. Admissible rules for Recommendation Interpreter

Following code snippet depicts the core logic where rules for food recommendation are differentiated from physical activity rules:



As illustrated in the aforementioned code snippet rules for food recommendation contain “*Food*” is key while rules for physical activity contains the keyword “*Recommended Activity*”. Please consult sample rules, provided in *Main.java* file, for creating new rules.

1. Running Scenario
   1. RI receives recommendation in the form of a production rule from Recommendation Builder (please refer to Recommendation Builder for the rule’s structure)
   2. *InterpretRecommendation* method in *RecInterpter* class serves as the starting point for RI processing
   3. Received input as in the form a *packet* (for the structure of packet please refer to *InterpreterPacket* class)
   4. Based on the contents of packet a decision is made whether to activate processing for physical activity recommendation or food recommendation (as mentioned at number 6)
   5. For example rule pertaining to physical activity is received
   6. *PARecInterpretations.java* is the key class which controls logic for processing different aspects (user’s interruptibility, contextual viability of the recommendation, explanation generation) of physical activity
   7. Final result of physical activity recommendation is stored in *InterpretedRecommendations* object
   8. *InterpretedRecommendations* object contains following key properties for physical activity recommendation
      * 1. Recommended activity (string)
        2. Recommended activity duration (string)
        3. Recommended activity description (string)
        4. Recommended activity URL (string)
        5. Recommended activity’s current context (string)
   9. For the case of food recommendation following two properties of *InterpretedRecommendations* are set:
      * 1. Recommended food item list (string)
        2. General description (which is part of the rule received from Recommendation Builder)
2. Recommendation Interpreter Architecture

Following architectural diagram depicts key components of RI. Context Interpreter deals with assessing the interruptibility of the user. Content Interpreter deals with assessing the contextual viability of the recommendation. SNS Trend Identifier specifically deals with food based recommendations. Explanation Manager provides explanatory note and audio/visual aids along with the recommendation. Data Manager provides local data repositories (data files).

