Unit 2: Case-based Reasoning (elective)

Prototyping Project: Recipe Recommender

*This project will contribute **30%** to the overall exam grade for this Unit. Note that you may fail this unit if you do not hand in this assignment or have not made a reasonable attempt to work on this assignment. There will be a penalty for late submissions (3% per working day).*

Objectives

The objectives of this prototyping project assignment are:

a. to reinforce the theory, principles and techniques of CBR taught in class;

b. to apply some of the guidelines given in class;

c. to illustrate some of the considerations that must be made in designing and building a CBR application, ie., applying the CBR development process. In doing so, the power and potential of CBR is highlighted, as well as the problems faced in building and testing a CBR application;

d. to provide a better understanding and appreciation of implementing a CBR system. You may use any available CBR tool or programming language. If you are using a tool, then discuss the essential features that a CBR tool must have.

Introduction

Thirsty? Enjoy a drink after a day of hard work! Your task in this CA is to build a Case-based Reasoning system that can help a user to DIY a delicious cocktail. Your system is required to create a suitable cocktail recipe given a query consisting of a few desired or undesired ingredients and other requirements for the drink.

Here are a few examples of possible queries:

- *I have martini and orange juice. What drink can I make?*
- *I want to prepare a drink using apple juice with no extra sugar added.*
- *I want a recipe for non-alcoholic drink using orange juice and lemon juice.*
- *Recommend me a simple cocktail recipe using at most two liquid ingredients.*

- *Any recipe for sweet cocktail with citrus juice for 2 persons?*
- *I want to prepare a "strong" drink (i.e. high percentage of alcohol).*

Some queries may be answered based on the available recipes by selecting a suitable recipe without modification. Others may require modification such as the exchange or removal of an ingredient.

The above example queries are in free text format. However, they can be manually transformed into a format suitable for the UI of your system. The answers from the system can be either a single or up to five recipes, including a note stating which original recipes from the library have been used to create the result.

You are given an xml file containing around 100 cocktail recipes. Study the recipes carefully and decide how you are going to use them in your system. You have the flexibility to decide how the cases are to be represented by identifying the salient features of each case. You may put more recipes into your case library.

As a guideline, before you begin implementing your CBR system, consider the following:

□ State clearly your goal of the system.

□ What is a case? That is, what are the essential information to represent in order to achieve your system goal; and what information must be stored and displayed for browsing purposes? For each case description, is it to be represented as 1 monolithic case, or do you break it into sub-cases? Is there any categorization or classification that must be done beforehand by the domain expert?

□ Case library design considerations - how to organize the cases? Assume that the final system may have thousands of cases stored.

□ Indexing vocabulary (through the *functional* approach or the *reminding* approach), and choosing good indexes for the problem solving task, taking into account efficiency and accuracy issues;

□ Decide whether to perform inductive or nearest neighbour retrieval (and give your reasons), or both.

□ How can you improve the performance of the retrievals; in terms of speed and accuracy?

□ What adaptation is to be carried out, hence what adaptation formulas to build?

□ Create the user interface and discuss how the system interacts with the user.

□ Testing - you have been given some example queries. Use these to test your system. What happens if there is no exact match for the type of cuisine/meal/ingredients? How does the system decide the next best? Create your own test cases if necessary.

In considering the above points, list down the assumptions that you make, regarding the culinary domain, as well as redundant, inaccurate or missing information, if applicable.

Instructions

1. For this project, you are to work in a **team of 4±1 students**. Create an external file of cases which can then be imported into your library.

2. Submit the following by **30 June 2017 (Friday) 6.00 pm**:

   a. a softcopy of Project Report
   b. the system implemented (including source code and data file)

   into your project team's "Files" folder, in "KE4202 Case Based Reasoning" on IVLE.

   The weights for the report and the system will be evenly distributed (i.e. 50% each). The report should be a critical discussion of important CBR issues that are demonstrated in the process of implementing the system.

   Include instructions for running your system in the report.

   Your report will be graded based on coverage and clarity, depth of discussion, logical flow and grammaticality. The instructions to run your system (or User Manual) should be clear, concise, and easy to understand.

   The system will be graded based on the innovativeness and complexity of your solution to the problem, the implementation efforts, and the **verification and validation** performed.

3. ***Suggested*** format for Project Report:

3.1 Problem Description and Objectives of the Recipe Recommender System

3.2 Problem solving paradigm - why is CBR appropriate? How does it compare with how an human expert might evaluate a recipe?

3.3 System Description:
3.3.1 Operational Context (profile of system's intended users, interaction between system and user, etc.)
3.3.2 Functional Description (functions performed or supported by the system)
3.3.3 System design considerations/architecture
3.3.4 Case-base Structure and representation. What information is stored and what is not? Why?
3.3.5 Assumptions (eg. relative importance of the case features) that you make.

3.4 Validation and verification:

3.4.1 Procedure & Results. Test your system with test cases (the given queries and other cases that can demonstrate what your system is capable of), and give the actual retrieval results in a table.

3.4.2 Discuss the results. What are the difficulties faced?

3.5 Findings and Recommendations:

3.5.1 Advantages and disadvantages of CBR (as applied to recipe recommendation) - eg. issues related to case-base maintenance, flexibility of system, efficiency of storage and retrieval, etc.

3.5.2 Alternative design/architecture? Hybrid system with rules, fuzzy logic??

3.5.3 If using a tool: discuss the strengths and weaknesses of the tool in handling this problem. Suggest additional functionalities that would be appropriate for this problem.

3.6 Conclusion/Future enhancements. Any lessons learnt or problems faced?

3.7 General Discussion. Eg. how ‘intelligent’ is the system in interpreting the given cases/queries? Is its judgement better than a human’s? What are the ‘real world’ problems (as in the actual cases given here) faced by an ‘artificial’ system? Besides using the system to suggest a recipe, assuming that you have enough cases and the needed information has been captured, what other potential uses can you think of? Use your imagination and have fun! ☺