Jonathan Louie
CS7637

Assignment 5

INTRO

Developing new recipes is as much of a challenge for the beginner chef as it is for the professionals. Considering new blends ingredients is always a challenge, on top of figuring out perfect cook times for ingredients that the chef has never used before. Using AI topics such as Version Spaces, and Constraint Propagation, an AI agent could be developed to aid the automated process of recipe creation, opening new doors for beginner and experienced chefs alike.

AI TOPIC EXPLANATION

Version Spaces is an incremental learning technique used to find a target concept model that has a balance between specificity and generality. Positive and negative training examples are used to create this target concept model. In our case, the model would be a recipe and its attributes would include the list of ingredients and the preparation instructions. A positive training example would be a recipe that tastes good while a negative training example would be a recipe that tastes bad.

Constraint Propagation is the characterization of a problem such that certain conditions must be met. These characterizations are called constraints. In this particular problem, constraints are imposed both by nature of the ingredients as well as by the chef. For example, a natural constraint might be the cook time of a chicken drumstick or a pork chop, while a user-imposed constraint would be the requirement of one or more ingredients or a total number of ingredients.

AGENT DESIGN

The agent would start off by being trained. The user would give the agent a number of recipes that he/she likes or dislikes. The agent would store these examples in a large decision tree, which would track which combinations of ingredients taste delicious to the user and which ones do not.

After training the AI agent, it is then ready to produce recipes based on user input. For example, the user input might look like:

[Chicken, tomatoes]

This would signify that the user wants a recipe requiring both chicken and tomatoes. The AI agent might notice that the user had previously enjoyed a recipe containing both chicken and thyme as ingredients. It would then check to see if there were any recipes containing thyme and the remaining ingredient (tomatoes) that were poorly received by the user. If not, then thyme could be selected as a third ingredient, in a newly produced stew. Based on the type of recipe that this is (a stew), the AI agent would produce cook time instructions appropriate for that of a stew (simmer all ingredients 1 hour or longer).

If the user chooses following the recipe, he/she could critique it and the results could be used as yet another training example to continue refining the agent.

CHALLENGES

There are certainly a few challenges and considerations that would come with designing an AI agent used to design new and delicious recipes.

- What is delicious to one person might be completely unappetizing to another. Thus, an AI agent developed with this implementation would need to be trained separately for each user. This means the learning portion of the AI Agent cannot use crowd-sourced data.
- By not being able to crowd-source data, it would take a very long time for this AI agent to reach full maturity. This is because full maturity would require many, many various training recipes to understand all of the different combinations of ingredients that the user likes or doesn't like.
- Because of the large number of potential ingredients and even larger number of ingredient combinations, the decision tree used to store the user's recipe and ingredient preferences would grow to be very large and sub-optimal due to the continuous learning process. The growth in complexity of this tree could cause performance issues.
- Including an algorithm to create an optimal tree from the current sub-optimal tree could minimize performance issues. This might be a necessity for the real world version of this implementation, but developing it would be a very difficult problem in itself.