

CO659 Project: Gourmake
Project Report

Chris Bailey
cb661@kent.ac.uk



School of Computing
University of Kent
United Kingdom

May 7, 2018

Abstract

This paper describes the development, operation, conclusion and my reflection on a computational creativity system which invents recipes, henceforth referred to as 'Gourmake'.

Gourmake is an attempt to declaratively generate creative artefacts. Domain knowledge is simply encoded and presented to the system which then probabalistically pieces them together in an appropriate fashion.

At the time of writing, the Gourmake system is arguably a successful one; having achieved its goal of being able to produce human-like output. Despite several failings in the implementation due to misjudgments and time constraints, Gourmake is currently able to generate what appear to be rather novel, inventive and legitimate recipes.

1 Introduction

The primary goal of Gourmake was originally to be able to create simple and human-like recipes drawing on no knowledge other than basic information about ingredients and methods of preparation — whether or not Gourmake has achieved this is up to interpretation but at the time of writing Gourmake can and does generally produce human-like recipes.

Gourmake does this by drawing inspiration from existing recipes, which are heavily templatised at varying levels of abstraction, allowing essentially for Gourmake to autonomously fill in the gaps by cross-referencing categorisations of different ingredients and finding suitable ones to use; which whilst simple, is very extensible and produces surprisingly original and 'thoughtful' results.

The templatised recipes which drive the Gourmake system are heavily declarative, and Gourmake represents a declarative approach to generative artefacts.

This paper goes on to describe the background research which challenged original ideas about how Gourmake ought to be designed, as well as key implementation details which help Gourmake successfully achieve its primary goals. This paper also discusses the limitations of Gourmake, how to improve Gourmake and the future of Gourmake.

2 Background

There is much discussion as to what the definition of 'Computational Creativity' really is. This definition, of course, is one which must be tackled as it ultimately informs the direction of this project, its goals and its outcome.

Wiggins, for example, describes 'Computational Creativity' as "behaviour exhibited by natural and artificial systems, which would be deemed creative if exhibited by humans" [Wiggins, G. 2006. A preliminary framework for description, analysis and comparison of creative systems]. In line with this, and other thoughts such as from [Cognition as part of computational creativity], "humans are the only arbiters of creativity", the ultimate goal of the Gourmake system is to create recipes which can be deemed human-like and otherwise indistinguishable from any average recipe.

Within the field of Computational Creativity, there have been many such systems in the same domain. All of these systems do ultimately share some similar ground and similar approaches which help lay the foundation for ideas further developed and used in Gourmake.

Many such systems used as research to inform the direction of Gourmake utilise essentially at their core probabilistic database searches iteratively through different methods, building up lists of ingredients. Of such systems, the most common elements shared between them were databases of many recipes and many ingredients. [Cognitive Cooking] [Cognition as part of computational creativity]

Gourmake functions in a way inspired by the PIERRE system [Soup over bean of pure joy], creating recipes in a layered abstracted approach and as a result can create novel, interesting but more importantly also rather edible artefacts.

At a high level, Gourmake, like the PIERRE system [Soup over bean of pure joy] creates dishes from an inspiring set of recipes, as well as many layers of abstraction for ingredients which

are to be used in these recipes. We will explore these abstractions and ideas underlying Gourmake’s generative process further in the paper. One of the goals of this project was to try and use a similar system to PIERRE[Soup over bean of pure joy] whilst not necessarily being limited to the scope of ”soups, stews and chillis”.

Starting this project, during planning, Gourmake was meant to be an evolutionary styled algorithm with a very similar fitness function that that described in the PIERRE[Soup over bean of pure joy] system whereby a criterion for evaluating a recipe’s success was its likeness in terms of the amounts of ingredients within specific categories to those of real recipes.

Retrospectively, whilst Gourmake doesn’t at the time of writing contain any evolutionary behaviour, the main limitation holding Gourmake back is the lack of a large normalised recipe set to both learn from and be compared to.

The idea of building recipes by abstracting the core components of said recipes into ’categories’ is interesting and evidently workable. Similar approaches are use by [Cognitive Cooking], [Cognition as a part of computational creativity] and [New developments in culinary computational creativity], although the complexity of a ’categorisation’ process is oft-times fairly complex. [New developments]’s computational model classifies ingredients into ’categories’ via ’Odor Similarity’ and other such means, which for Gourmake, isn’t something to be feasibly implemented due to the fact that Gourmake’s databases and learning set are hand generated.

The ingredient database available to similar systems also oft-times encodes additional information as well such as ”Cultural Context” of ”chemical analyses of ingredients” as well, such as in [Cognition as part of computational creativity]. Again, because of limitations in the resources afforded to the Gourmake system, Gourmake relies on the quality of the human-encoded knowledge as part of its ’Inspiring Set’ of recipe templates to produce high-quality output.

Other means to arguably increases the apparent originality or creativity of a system are implemented by [New Developments in Culinary Computational Creativity]’s system which can consider autonomously substituting ingredients for one another (in their case, from a given meat based product to a vegetable). Having been inspired by this, the creativity of Gourmake arises from partially from the sheer unpredictability of Gourmake’s output due to probabalistic noise. Gourmake actively makes decisions to differ from the learning set, which at times produces surprisingly ”creative” output.

3 Methodoly and Design

3.1 Implementation and design of Gourmake

Gourmake was designed as a multi-agent system consisting of several autonomous agents, all of which play certain creative roles which come together to produce Gourmake’s artefacts. This design was chosen primarily to abstract away different procedures in recipe construction and to be easily extensible. More agents which provide certain functionality can easily be added and plugged into the creative process if needed.

Gourmake is written in Erlang and naturally utilises Erlang’s underlying Actor Model message passing mechanisms to communicate between individual agents. This also allows us several

benefits such as easy scalability and concurrency support making Gourmake ideal for a web-app or web-frontend allowing people to easily request new recipes on the fly.

Gourmake is at a high level, composed primarily of two agents, one agent specialises in parsing our recipe database and finding correlations and categorisations between ingredients allowing us to query for any number of random ingredients given arbitrary constraints. The other agent specialises in understanding the recipe database fed into Gourmake, and essentially coordinates different agents together to be able to produce output.

3.1.1 Ingredient Database Agent

The Ingredient Database Agent is, as stated, responsible for handling, choosing and understanding individual recipes in Gourmake's creative process.

Gourmake is, at the time of writing, pre-loaded with knowledge of 629 ingredients (datamined from the BBC[1]) which is contained in a plaintext file read in by Gourmake. The file is essentially an Erlang term which is evaluated at runtime and works essentially the same as JSON. This datastructure maps a given ingredient name to arbitrarily many different categories and cuisines associated with said ingredient which the Ingredient Database Agent processes to learn.

The categories and cuisines which are associated with individual Ingredients are represented by Erlang atoms and thus are only data labels which means adding new cuisines or categories to a given ingredient is as simple as modifying the file read in by Gourmake, nothing else needs to be done.

Because of this, any category or cuisine can be added allowing Gourmake the implicit ability to abstract the categorisation of its ingredient data to an arbitrary level. Ingredients can be categorised however a given individual wants, be it as simple as *"A root vegetable"* to *"Something spicy"*.

Because boolean logic can be applied to categories and cuisines, unlike the PIERRE system, I've found it unnecessary to nest categories because a similar effect can be constructed to their examples by simply finding the union or intersection of the results of multiple categories.

Once this file has been parsed, the Ingredient Database Agent processes it to build an 'network' representation of all of the ingredients. This network subdivides individual ingredients into lookup tables for categories, cuisines and ingredient names which gives the Ingredient Database Agent the ability to quickly and efficiently search through our list of processed ingredients cusinewise, categorywise or ingredient namewise.

When other agents communicate with the Ingredient Database Agent, they can then provide constraints such as *"An aromatic spice, which isn't sweet nor a dessert, typically used in indian cuisine"*. After a series of filters and searches through our network, the Ingredient Database Agent returns one ingredient which best matches the given constraints.

If a constraint cannot be satisfied, the Ingredient Database Agent expands its search for similar categories, cuisines or simply weakens the constraints placed upon it until it successfully finds an ingredient. There is also a stochastic probability that the Ingredient database agent will ignore constraints or invent new ones, which results in less predicatble output.

There was more work to be done on the Ingredient Database agent, but this is discussed further down.

3.1.2 Recipe Database Agent

The Recipe Database Agent is at a high level not dissimilar to the Ingredient Database Agent, except it does processing on a file containing an Erlang Term representing our 'Learning Set'.

This 'Learning Set' is a set of templatised recipes which encode human knowledge ready to be filled in. Currently, at the time of writing, the 'Learning Set' consists only of 16 examples. This was simply due to the fact that all of the datasets used in Gourmake are hand written to conform to the exact scheme created for Gourmake. However, even with only 16 example recipes in the 'Learning Set', the output of Gourmake always seems rather original and there is seldom much similarity between generated recipes.

The formatting of the 'Learning Set' is again, JSON-like, and a particular recipe encodes information about how it should be named, what abstract categories of ingredients should be used (as well as minimum and maximum quantities of each category, and constraints imposed upon a pending Ingredient-Database query) and valid cuisines a recipe can be tweaked to satisfy.

The Recipe Database Agent simply parses through the encoded template recipes, and makes decisions such as how many ingredients of a given category need to be selected and used for this recipe. That means that recipes vary not only in what ingredients are used but how many of each ingredient category are present. Gourmake does not yet provide any measurements for ingredients as there is no normalised weight system implemented in the Recipe or Ingredient databases yet but this was pending being added.

The Recipe Database Agent ultimately is the component which does all of the work; it communicates with agents such as the Naming Agent (which is responsible for naming recipes based on simply looking at an underlying recipe and selected ingredients), and the Ingredient Database Agent, coordinating them to produce an artefact.

3.1.3 Miscellaneous Components

Gourmake also utilises some miscellaneous components to improve the running of the system. All of our server agents are supervised by a higher level supervisor which can react to individual servers crashing. The supervisor automatically restarts such processes and ensures everything is running well.

There exists also a Naming Agent which names recipes based on the template provided by the recipe as well as the generated ingredient list produced by the Recipe Database Agent. For more personality and character, the Naming Agent also randomly selects a noun or adjective to prepend onto the generated recipe name allowing for names such as *"Ritchie's European inspired One-pan Farfalle with Beef Ribs, Radish and Sweetcorn"* or *"Knuth's Simple Emmental and Gruyere omelette"*.

3.2 Remaining work to be done

Despite the fact that Gourmake can successfully produce human-like output, and create at times rather sensible, original and inventive recipes, due to time constraints, much work which was intended to be complete isn't at the time of writing complete. The following features were planned to be implemented and will be implemented at a future time to improve Gourmake's generated artefacts.

3.2.1 Abstraction Layer for Ingredient Preparation

At the time of writing, Gourmake can abstract details about cuisine and categories of specific ingredients to allow it to select random yet 'thoughtful' ingredients for recipes, however, each step of a recipe is currently hard coded in the recipe templates which exist in our 'Learning Set'. This means that the individual steps for each recipe are fixed and cannot change.

For more varied output, Gourmake would need to be able to abstract how to prepare certain ingredients and stochastically choose preparation methods based on cuisine and ingredients used rather than hard coding. This isn't a particular difficult task to implement, it just has not been implemented yet due to time constraints. Much like the Ingredient Database Agent, a Preparation Database Agent would need to be added which would encode information about how particular ingredients are best prepared, commonly prepared and have these also substituted into a recipe template.

3.2.2 Self-Evaluation of Produced Artefacts

Gourmake was originally going to use a method of generation based on Evolutionary Algorithms where the produced output of Gourmake would only be presented to a user after many generations of 'evolving'. The main issue with self-evaluation is finding a means of self-evaluating.

A proposed fitness function was to compare the similarity of chosen ingredients to those in other recipes, or to have real world example recipes from which recipes in the learning set are based off, much like in [Cognition as a part of computational creativity]. The issue with this and quite literally the only reason this hasn't been done is because the learning set is currently so small.

Once this is implemented, we can gradually evolve more realistic looking recipes over time which would only increase the quality of Gourmake's output.

3.2.3 Improvements to Ingredient Database Agent

Utilising Self-Evaluation and having access to real-world recipes to learn from, a proposed change to the Ingredient Database agent is an improved probability to select ingredients which are used alongside already selected ingredients. Currently this isn't done and instead is implemented such that there is a low probability of substituting any ingredient with any ingredient in an adjacent category to the proposed ingredient.

4 Results

Despite the fact that many planned features were not implemented due to time constraints, Gourmake still arguably produces human-like artefacts which at times can appear to be quite creative and novel. Some raw examples of Gourmake’s output are below (*more examples can be found on Gourmake’s GitHub repo at <https://github.com/vereis/gourmake>*):

```
# Gosling’s Asian inspired Cod and New Potato Casserole with Radicchio and
  Creamed Coconut and Tartare Sauce
## Ingredients:
- New Potato
- Cod
- Radicchio
- Creamed Coconut
- Tartare Sauce
## Instructions:
1) Preheat oven to around 300 - 350 degrees.
2) Combine Cod and Radicchio in a mixing bowl and mix until even.
3) Then, add your Creamed Coconut and incorporate it into the contents of your
  mixing bowl
4) Move your mixed ingredients into a pot and cook on the stove until cooked
  through.
5) Finally, move your mixed ingredients into a baking tray and top it with your
  Creamed Coconut and Tartare Sauce.
6) Bake the contents of your baking tray for 20-30 minutes or until golden
  brown and enjoy.
```

You can clearly see that the generated name for the dish successfully sounds human-like and not necessarily generated outside of the clumsy grammar when it comes to conjoining multiple items. In this particular case, the ingredients chosen successfully reflect what seems to be a valid and legitimate recipe.

Everything except the instructional text, which simply acts as glue, is stochastically generated and is the result of nothing but probability manipulation as defined by the Ingredient Database Agent’s ingredient network as well as the Recipe Database Agent’s probabalistically selected cuisines, ingredient categories and other constraints.

5 Evaluation

6 Conclusion

Retrospectively, whilst the Gourmake system does produce what is in my opinion rather human-like and at times surprisingly inventive recipes, the Gourmake system suffers from the fact that the learning set it has to utilise isn't very large and thus, despite the fact that the generated permutations of the learning set are vast, patterns which come from the raw plaintext encoded in the learning set are easy to see.

As stated, with more time and more resources, Gourmake would ideally be improved by simply encoding more recipes into Gourmake's learning set as well as encoding other recipes in their raw form as a sort of evaluative set so that Gourmake can iteratively not only build recipes stochastically but also evolve them to meet certain criterion; however, the approach Gourmake uses by abstractly categorising ingredients at a high level is seemingly a successful one.