Smart Chef: Evolving Recipes

Carsten Draschner, Jens Lehmann, Hajira Jabeen
{draschne,jabeen,jens.lehmann}@cs.uni-bonn.de
University of Bonn, Germany

ABSTRACT

Smart Chef demonstrates the creativity of evolution in culinary arts by autonomously evolving novel and human readable recipes. The evolutionary algorithm for Smart Chef fully automatized and does not require human feedback. The tree representation of recipes is inspired by genetic programming and is enriched with semantic annotations extracted from known recipes. The fitness identifies valid recipes and novelty. Recipe mutation exchanges ingredients by food category classification and recombination interchanges partial recipe instructions. Smart Chef has been tested on a population size of 128 and evolved for 100 generations resulting in valid and novel recipes.

KEYWORDS

evolutionary algorithm, artificial creativity, recipe, culinary, semantic creativity, genetic programming, food graph, recipe annotation, human readable recipe representation

1 INTRODUCTION

Computational creativity is an emerging branch of artificial intelligence that places computers in the center of the creative process. The recently published approaches are focused on selected domains like Graphics or Music generation. Food is an essential part of our life and the dishes we eat are created using various recipes. These recipes demonstrate creativity in combination of ingredients, methods, tastes, textures and proportions. Smart Chef presents an automated system capable of creating novel human readable recipes using Evolutionary Algorithm(EA) for recombining recipes from different regional cuisines.

2 RECIPE AS TREES - MACHINE READABLE DATA REPRESENTATION

The initial recipes are fetched from theMealDB.com [1] JSON API. The phenotype to genotype mapping is semi automated. It creates a semantic annotated tree structure inspired by genetic programming (see Figure 1, 2.A, 2.B). The root node represents the final recipe. Each inner node is a granular task (grey) manually constructed based on the preparation sentences. These nodes are assigned a *node-type* and *instruction-type* and attributes, if they are mentioned in the sentence (see Fig. 1). The child nodes of instruction-nodes are either inner instruction nodes (grey) or leaves (blue) identified in the sentence. The leaves are ingredients which have a *name* and a *proportion* and are automatically generated from the recipes ingredient table (see figure 1).

Proceedings in ArXiv of EVO 2019 - Late-Breaking Abstracts.*

3 RECIPE GENERATION USING EVOLUTIONARY ALGORITHM

The evolutionary algorithm generates new recipes in each generation. All the steps of the EA are designed to be fully autonomous for an arbitrary population size.

INITIALIZATION: The initial population consists of 128 pre-processed annotated recipe-instruction-trees(see Section 2 and Fig. 1) from theMealDB.com[1]. This population is further evolved to discover novel recipes.

FITNESS EVALUATION: The fitness of each recipe is calculated automatically. For this purpose, multiple characteristics of the recipe are taken into account and compared to known recipes. These characteristics are extracted from known valid theMealDB recipes. The recipe normality regarding number of ingredients and procedure steps (related to effort) from the novel recipe are compared to known recipes. The ingredient composition (main-, side-ingredients, spices) patterns are compared the common pattern from known recipes. The children recipes might use same ingredients multiple times in same recipe which is punished in the fitness function. The ingredient-set similarity compared with known recipes is used to give new ingredient combinations a higher creativity score. All those criteria extracted from recipes go into a weighted sum which defines fitness value of each recipe.

SELECTION We have used tournament selection based on the assigned fitness value for fitness evaluation.

RECOMBINATION: The crossover combines two recipe trees. From the one parent recipe tree a random subtree is replaced by a subtree from the second recipe with similar characteristics (same sub-tree-size). These novel recipes create the next generation of EA for evolution.

MUTATION: In each child recipe, one ingredient is replaced by an ingredient from food-databases[2, 3]. The chance for each ingredient in this database to be used is dependent on the food similarity (see Fig.2.D) based on its hierarchical classification (i.e. in foodsubs[3] e.g. Spaghetti has hierarchical classification: Pasta-Rods, Pasta, Grain-Products, Food).

RESULTING POPULATION The EA runs for a certain number of generations. For the final population a fully automated genotype - phenotype mapping creates a human readable recipe descriptions with a recipe title, an ingredient table with proportions and fulltext-instructions in natural language (see Figure 1)

4 CONCLUSION

This work shows that novel recipes can be fully automatically generated from a genetic programming inspired approach. The proposed genotype-phenotype mapping creates common recipe structures from the recipe trees. For future work, this concept allows arbitrary extensions for fitness evaluation (e.g. recipe: price, sustainability, diet/heath etc.).

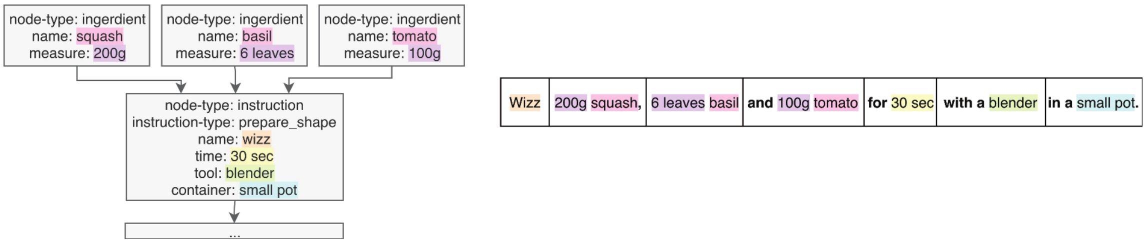

Figure 1: Example sentence for instruction representations of a recipe. annotated tree structure (left), human readable sentence (right). The phenotype genotype mapping is semi-automatized. The genotype phenotype mapping is fully automatized.

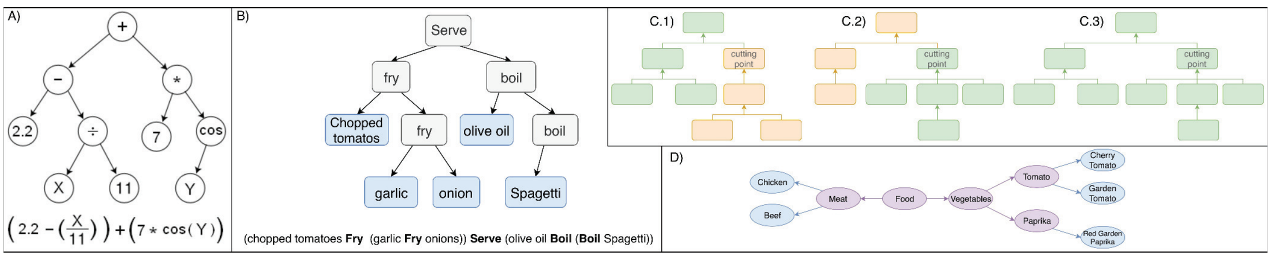

Figure 2: Genetic Programming Example (A), Corresponding simplified Recipe Example (B), Recombination of Trees (C), Ingredient Hierarchy Example (D)

REFERENCES

[1] Alias: zag@kodi.tv. themealdb. https://www.themealdb.com/, 2016-2019. Accessed: 2019-03-10.

[2] The Metabolomics Innovation Centre (TMIC). foodb.ca. foodb.ca, 2017. Accessed: 2018-01-21.

[3] Lori Alden. Foodsubs - the cook's thesaurus. http://www.foodsubs.com/, 1996-2006. Accessed: 2018-01-21.