Smart Chef - Evolving Recipes

Carsten Felix Draschner

Matriculation number: 2719095

July 11, 2019

Master Thesis

Computer Science

Supervisors:

Dr. Hajira Jabeen, Prof. Dr. Jens Lehmann

Smart Data Analytics - Institut für Informatik III

Rheinische Friedrich-Wilhelms-Universität Bonn

Declaration of Authorship

I, Carsten Felix Draschner, declare that this thesis, titled “Smart Chef - Evolving Recipes”, and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. Except for such quotations, this thesis is entirely my own work. I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: ____________________

Date: ____________________

Acknowledgements

I would first like to thank my thesis advisor Dr. Hajira Jabeen. She consistently allowed this paper to be my own work, but steered me in the right the direction whenever she thought I needed it. I am very thankful to Prof. Dr. Jens Lehmann for allowing me to pursue the master thesis in the Smart Data Analytics (SDA) research group that is a part of Institute for Computer Science at the University of Bonn. In the Cooperation with the Smart Data Analytics Group, this Master Thesis approach was published as a Poster (see figure A.11) at the Conference EvoStar 2019. I would also like to thank the participants who were involved in the validation survey for this research project. Finally, I must express my very profound gratitude to my parents and to my friends for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Contents

List of Figures

Abstract

The internet and cooking books provide millions of different recipes that describe the process to create the dishes we have and want eat every day. The choice of recipes is based on a multitude of criteria. The recipes effort and the underlying ingredient criteria like availability, price, sustainability and animal ethics are criteria which are important as the taste and coverage of daily needed nutrition. An autonomous creation of recipes can create recipes based on personal preferences and priorities. The exploratory behavior of evolutionary algorithms is used to discover novel recipes. The specific application of tree based genetic programming makes a semantic preserving data representation of the common recipe instruction texts possible. An initial set of known valid recipes are used to create novel recipes. In the recombination partial recipe instruction trees are merged and small mutations of the used ingredients result in creative recipes. These created recipes are filtered in an external selection based on assigned fitness values. The fitness value is automatically generated and includes information about the recipes novelty and validity. Also mentioned criteria like the effort and number of used ingredients are taken into account. Multiple Evolutionary Algorithm Cycles creates novel recipes which are autonomously transferred to human readable recipes in a Post-Processing. A meta data Analysis and an online survey shown that the autonomous recipe creation process works successful.

Keywords: Food, Recipes, Evolutionary Algorithm, Genetic Programming, Artificial Creativity

Chapter 1

Introduction

Artificial Intelligence became more and more a buzzword in computer science as well as in popular science and arts. There is a broader discussion about what are the capabilities of current artificial intelligence approaches. The potential of Artificial Creativity is very controversial [2]. Artificial Creativity means that a computer is capable of creating a new valid entity of the desired domain [3]. Novelty means that this newly created entity differs significantly from the known entities. Valid means that the entity serves a natural appearance, that fits roughly common expectations or purposes for members of the desired domain.

Food is a daily present and fundamental part of our lives. We need food to meet the daily nutritional needs of our bodies and to live that healthy. We consume food also because its taste can be pleasant and entertaining. The food we consume often results from recipes. Recipes describe how a set of ingredients has to be prepared to end up in the desired dish. The ingredients are precisely named and have a measure specifying how much of each ingredient is needed. On the internet, cooking websites present thousands of dishes. For each dish, sometimes hundreds of recipes describing the "best" way of cooking it. The alternative recipes differ in procedure and ingredient sets, which result in differences of procedures effort, ingredients price, ingredients availability, dishes healthiness or taste.

1.1 Motivation

Recipe creation through computer algorithms can optimize recipes under a multitude of desired criteria. On the one hand, the ingredient list can be used to calculate several characteristics of the resulting dish. The ingredients of a created recipe can automatically be evaluated regarding the daily needed nutritions which correspond to the healthiness of the resulting dish. Different ingredients are produced and used in different parts of our world. This can leads to differences in availability. The procedure to gain the ingredients and transport them to the kitchen has a CO2 or water footprint. So each ingredient has its sustainability characteristics. The ingredients differ in price, which results in different total costs for the created dish. The procedure of a dish defines the effort and the complexity of needed skills or experience. All these criteria can automatically be evaluated if the underlying data is given. An automatic recipe creation can consider all these criteria weighted by personal preferences in an optimization process. Computers can evaluate many more recipes than humans at the same time, so an early exploration and optimization of recipes serve a reasonable selection of exciting recipes.

1.2 Objectives

This Master Thesis presents the development of a pipeline that is capable of creating novel creative recipes by a computer. On the one hand, known valid recipes and information about ingredients are transformed into a semantic preserving machine-readable data representation. The recipe data representation is optimized to allow the application of the Genetic Programming procedure. The complete creation process runs automatically. The process creates novel recipes by combining known valid recipes and exchanging ingredients. Evolutionary Algorithms optimizes recipes based on their fitness value. A fitness function automatically calculates for each recipe a fitness value. This fitness value represents Creativity based on the novelty and validity of the recipe. For the automatic evaluation, data sets of food and recipes store the underlying knowledge of validity and novelty. The novel created recipes are presented in a traditional layout that is automatically created in a preprocessing based on the internal recipe representation. An online survey gives real human feedback for the validity and novelty of recipes.

1.3 Challenges

The development of a complete pipeline, for the autonomous creation of novel recipes, has many challenges. The creation of recipes by an Evolutionary Algorithm needs machine-readable recipes that fit the EA key conceptual steps.

Recipe Data Representation Usually, recipe instructions are presented in a linear text. This text names the ingredients and applied tasks. From the semantics of the linear text, parallel steps are understandable. Often several intermediate recipe results have to be prepared in parallel and combined in a later step. An example would be the parallel preparation of a sauce and cooked pasta which are combined in a further step before it gets covered by grated cheese. This semantic structure has to be understandable to the algorithm for reasonable handling of the described tasks and their ordering. Besides, recipes instructions name intermediate results by synonyms or descriptions. For example, the heated mixture of chopped tomatoes fried garlic and cubed onions are named "sauce" in further steps without explicitly writing this synonym assignment. Including this semantic knowledge is very challenging or needs a proper data representation that circumvents this uncertainty.

Recombination The Evolutionary Algorithm creates novel recipes based on the mixing of good recipes instructions. One part of recipe instructions are replaced with instructions from another recipe. It is not trivial to substitute an arbitrary part of instructions in a recipe. This combination of recipes needs a machine-readable data representation that robustly preserves the semantics. After such a substitution of a partial recipe in the internal semantic preserving data representation, a Postprocessing has to create understandable textual instructions.

Mutation The substitution of ingredients is the second step to changes recipes. Recipes are very sensitive to ingredient changes. An optimized substitution method of ingredients that replaces ingredients by reasonable substitution needs the underlying knowledge about ingredients similarity and characteristics.

Fitness Evaluation A fitness value represents the quality, novelty, or validity of a recipe. Usually, we tend to measure the fitness of a recipe by the resulting dish's taste. The measuring of taste would need the effort of cooking and trying out every generated recipe. Each dish has to be evaluated by many participants to reduce biased results regarding personal preferences. Evolutionary Algorithm changes recipe candidates over multiple cycles in Recombination (Mixing of recipe instructions) and Mutation (substitution of ingredients) steps. An evaluation of all intermediate recipes is not feasible. This approach needs an autonomous fitness function. This fitness function can only calculate and evaluate a few recipe criteria. In general, we expect that the instructions of a novel created recipe are understandable. The instructions need to be doable. Doable instructions mean that the procedure steps fit the proposed ingredients. A "boiling" step can only be applied if the set of ingredients is fluid enough. The resulting dish has to be at least edible and preferable tasty. It is desired that further criteria like the effort, the ingredients price, ingredients availability, and personal preferences are covered. When our aim is artificial Creativity, the fitness value needs to include novelty of the created recipes.

The pleasantness of a resulting dish is based on its taste, its texture, and its smell. The ingredients change their original taste and texture in preparation steps like heating. Heating changes ingredients in chemical reactions. An autonomous evaluation of intermediate recipe results needs deep chemical knowledge and simulation. In addition, the taste is hardly dependent on personal preferences.

1.4 Thesis Structure

The chapter Related Work outlines digital projects in the food and artificial creativity domain. The chapter Machine Readable Recipe Data Set presents the acquisition process of essential needed recipe and ingredient datasets. Also, the transformation from full-text instruction to tree-based genetic programming inspired data representation are mentioned in section 3.3. Evolutionary Algorithms applied to Recipe Creation describes the whole Evolutionary Algorithm process that generates novel recipes in a cycle process. The Post-Processing section presents how the internal tree structures are transformed into a human-readable recipe. Section Evaluation and Discussion presents results and metadata of the Recipe Creation Process. Section 5.2 presents the human feedback to created novel recipes gained from an online survey. An overview of further features and optimization potential are outlined in chapter 6. The final Conclusion of this Master Thesis Project is presented in chapter 6.

Chapter 2

Related Work

In the literature, no current project presents the capability of creating novel completely reasonable recipes autonomously. One group of scientific work [3, 4, 5] investigates which ingredients work together. These analyses partially deduce by considering the chemical level[6]. This analysis provides suitable ingredient combinations. The second group of projects develops machine learning algorithm with the ability of producing creative output [7, 6, 8]. The third group of scientific publications and business projects are focused on providing recipe and food data in a machine-readable format [4, 9, 10, 11, 12]. These machine-readable data representations try to preserve recipe semantics with annotations of the different recipe components.

2.1 Food Data Science and Recipe Analysis

In literature, some approaches investigate recipe characteristics. These approaches can be used to identify patterns in the choice for the right ingredient combination or to predict the recipe fitness based on underlying preferences.

Flavor network and the principles of food pairing This work uses graph theoretical approaches to investigate the similarity of ingredients based on the underlying flavor compounds. [5]. In this approach, they construct a weighted undirected graph. In this graph, nodes represent the ingredients. The edges have a weight that correlates to the similarity of ingredients sharing compounds. In their work, they observe differences in the choice of ingredients between different regions of the world.

Recipe recommendation using ingredient networks Teng, Lin, and Adamic (2012)[3] developed an algorithm, that is capable of predicting which recipe is better. This model uses network analysis and machine learning methods. The results are two ingredient graphs. One graph represents the co-occurrences of ingredients in recipes. Nodes represented the ingredients and edges represented the co-occurrence likelihoods. The second graph builds an ingredient substitution network. Nodes represent ingredients. If another ingredient could replace one ingredient, an edge connects both that indicates the substitution opportunity. This approach has an accuracy of 79% to predict the recipe to be the higher-rated recipe compared to another recipe.

2.2 Artificial Creativity

Artificial Creativity approaches that produce creative output and run autonomously on machines use various Computer Science methods. Many creativity domains are explored. Music have been composed[13], New Images are created[7] and sold[2]. Much fewer publications present results in the domain cooking, recipes or culinarian. All approaches have in common that machines should create creative output. A comparison of the novel produced output with valid entities of this domain is used for the evaluation if the novel creates elements are creative [7]. On the one hand, the newly generated objects have to be valid members of the domain (can humans distinguish which are new entities or do they fulfill the specified purpose). If known entities and novel generated have the same level of validity, the creation process delivers valid entities. The generation of those new entities is judged to be creative if these entities are sufficiently far away from being a copy[7].

Computational Creativity in the Culinary Arts is a project [14] that generates novel salad recipes with limited human feedback. Original salad recipes run through a statistical model to rank recipes. In another step, they experiment with various search algorithms to explore the salad recipe space and discover novel ingredient combinations. The ingredient combination selection is based on the work of Ahn et al. (2011)[5]. The novelty is evaluated based on the ingredient set. The resulting dishes don't include proportions and instructions.

IBM Chef Watson uses big data to create new recipes. Evaluation of foods chemistry level predicts suitable ingredient combinations. More than 10.000 recipes[8] from Bon Appétit are analyzed. Ingredient co-occurrences and the underlying sharing compounds are the key features for evaluating which ingredients fit together[6]. The novelty of newly created recipes is measured by comparing the sets of ingredients versus the ingredient sets of known recipes. The taste of a dish is approximated by the flavor giving molecules of the ingredients. The recipes are optimized under the criteria novelty, taste, and ingredient set. Also, users can enforce specific ingredients as a starting point. The results need to be interpreted because some instructions aren't clear and understandable enough [15]. Their work is not available anymore[1]. Only some blogs allow a few insights.

Evolutionary Meal Management Algorithm - Cover:Cheese is a project from Hampshire College Mad Science Club that implements an evolutionary algorithm that generates novel recipes. The optimization is done based on the taste rating. They crowdsourced their created recipes to gain feedback[16], which built the recipe fitness value (taste-rating). The generated recipes are not easily understandable or don't make sense at all. All of them are very short. Also, some recipes are pretty trivial[2].

2.3 Machine Readable Recipe Data Set

For the data analysis of recipes, the recipe data must be available in a digital format. Recipes include a title a list of ingredients (with the proportions) and instructions (how to prepare the dish). If a human reads a recipe in a cooking book or a website, layout guides to identify the context of presented text elements like an ingredient list and the semantics of the instructions. For data science methods, it is helpful that the contained data is annotated. The central information, like ingredients, title, preparation steps, and proportions, should be annotated to compare and analyze different dimensions of recipes easily. The more annotations are available, the easier machine readable the recipe representation is.

It is interesting for this project how other projects and services provide recipes. Is it necessary to annotate own data sets or transfer cooking book content into annotated data manually or are useful solutions available to get an annotated recipe data set? If such a recipe data set is available, how do the implemented structure and data representations of other projects and services can be used? Some suggestions of recipe structure recommendations are available [17], some recipe data sets which are not annotated [18],[19] and some services which provide a JSON API that serves recipes in a dictionary format. Some of the API services are for free [12] and some are professional charged services [11],[10],[20]. They have differences in the information they

[1]http://www.ibmchefwatson.com

[2]https://covercheese.appspot.com

offer and the annotations (granularity) they provide. Also different annotation patterns are possible [21]. Every book or website has a unique structure for recipe annotations.

schema.org - Recipe Schema This website recommends a structure for recipe annotations[17]. This schema is in a dictionary-style (key-value pairs). This schema has the main group *Properties from Recipe*. This group includes information about the final dish and some overall information about the instructions.

Properties from Recipe		
`cookTime`	Duration	The time it takes to actually cook the dish, in ISO 8601 duration format.
`cookingMethod`	Text	The method of cooking, such as Frying, Steaming, ...
`nutrition`	NutritionInformation	Nutrition information about the recipe or menu item.
`recipeCategory`	Text	The category of the recipe—for example, appetizer, entree, etc.
`recipeCuisine`	Text	The cuisine of the recipe (for example, French or Ethiopian).
`recipeIngredient`	Text	A single ingredient used in the recipe, e.g. sugar, flour or garlic. Supersedes ingredients.
`recipeInstructions`	CreativeWork or ItemList or Text	A step in making the recipe, in the form of a single item (document, video, etc.) or an ordered list with HowToStep and/or HowToSection items.
`recipeYield`	QuantitativeValue or Text	The quantity produced by the recipe (for example, number of people served, number of servings, etc).
`suitableForDiet`	RestrictedDiet	Indicates a dietary restriction or guideline for which this recipe or menu item is suitable, e.g. diabetic, halal etc.

Figure 2.1: schema.org recipe schema, Properties of Recipe

The instructions can be a list of *HowTo* information. This *HowTo* information includes much fine granular information of one preparation step. The example recipe on schema.org provides the instruction as a text. The website schema.org provides only one example to present the schema and not a set of different recipes. Currently, there is no data set available which follows this schema in total.

An Ontology Design Pattern for Cooking Recipes – Classroom Created The paper presents a description and result of an ontology modeling process (see figure 2.3) taken to the classroom. The application domain is recipes. The modeling goal is to bridge heterogeneity across representational choices by developing a content ontology design pattern, which is general enough to allow for the integration of information from different web sites. Various recipe sources have different recipe representations [21].

```
{
  "@context": "http://schema.org",
  "@type": "Recipe",
  "author": "John Smith",
  "cookTime": "PT1H",
  "datePublished": "2009-05-08",
  "description": "This classic banana bread recipe comes from my mom --
the walnuts add a nice texture and flavor to the banana bread.",
  "image": "bananabread.jpg",
  "recipeIngredient": [
    "3 or 4 ripe bananas, smashed",
    "1 egg",
    "3/4 cup of sugar"
  ],
  "interactionStatistic": {
    "@type": "InteractionCounter",
    "interactionType": "http://schema.org/Comment",
    "userInteractionCount": "140"
  }
```

```
  "interactionStatistic": {
    "@type": "InteractionCounter",
    "interactionType": "http://schema.org/Comment",
    "userInteractionCount": "140"
  },
  "name": "Mom's World Famous Banana Bread",
  "nutrition": {
    "@type": "NutritionInformation",
    "calories": "240 calories",
    "fatContent": "9 grams fat"
  },
  "prepTime": "PT15M",
  "recipeInstructions": "Preheat the oven to 350 degrees. Mix in the
ingredients in a bowl. Add the flour last. Pour the mixture into a loaf
pan and bake for one hour.",
  "recipeYield": "1 loaf",
  "suitableForDiet": "http://schema.org/LowFatDiet"
}
```


Figure 2.2: schema.org recipe example in a JSON dictionary format. split top half(left) and bottom half(right).

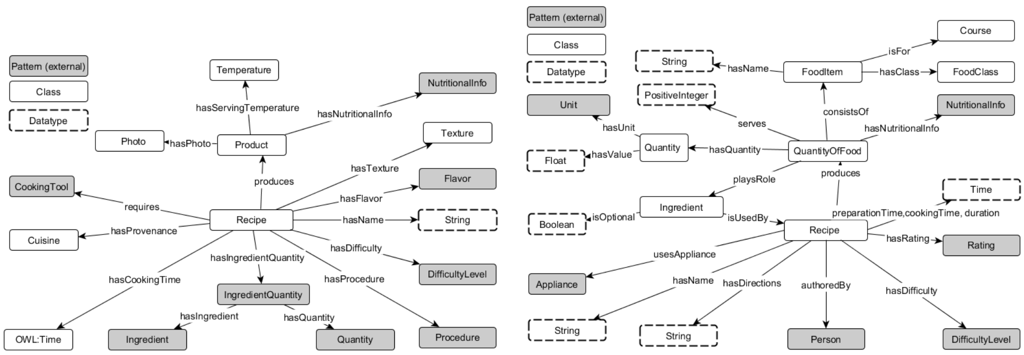


Figure 2.3: Two recipe annotation patterns preserving recipe semantic structure for computer readable recipes.

The patterns are designed to allow semantic questions for the characteristics of recipes in a recipe ontology. In their evaluation, they see much standard information across different recipe databases is annotated similarly. The approaches do not include a detailed representation of recipe instruction representation.

theMealDB.com is a website which provides a free[3] use of recipes through a JSON API [12]. The set has a size of almost 200 recipes (March 2019). Each recipe can be requested over a JSON API[4]. The JSON has 50 key value pairs[5]. Twenty are for ingredient name, and 20 are for ingredient proportion. These keys are always set. If a meal has less than 20 ingredients, the values are set to an empty string or null value (see figure 2.4. The other keys are for the recipe title, the area (26 different countries) and the category the recipe is associated with. The instructions are represented in a text.

[3] https://www.themealdb.com/api.php
[4] https://www.themealdb.com/api.php
[5] https://www.themealdb.com/api/json/v1/1/search.php?s=Arrabiata

The ingredient proportions aren't standardized. Often the less standardized units like tablespoons, teaspoons or pinch are used to indicate the proportion. Also, the numbers of ingredient elements are used to describe proportions. Because food has no standardizes size, a transformation into standardized measures like weight or volume is difficult.

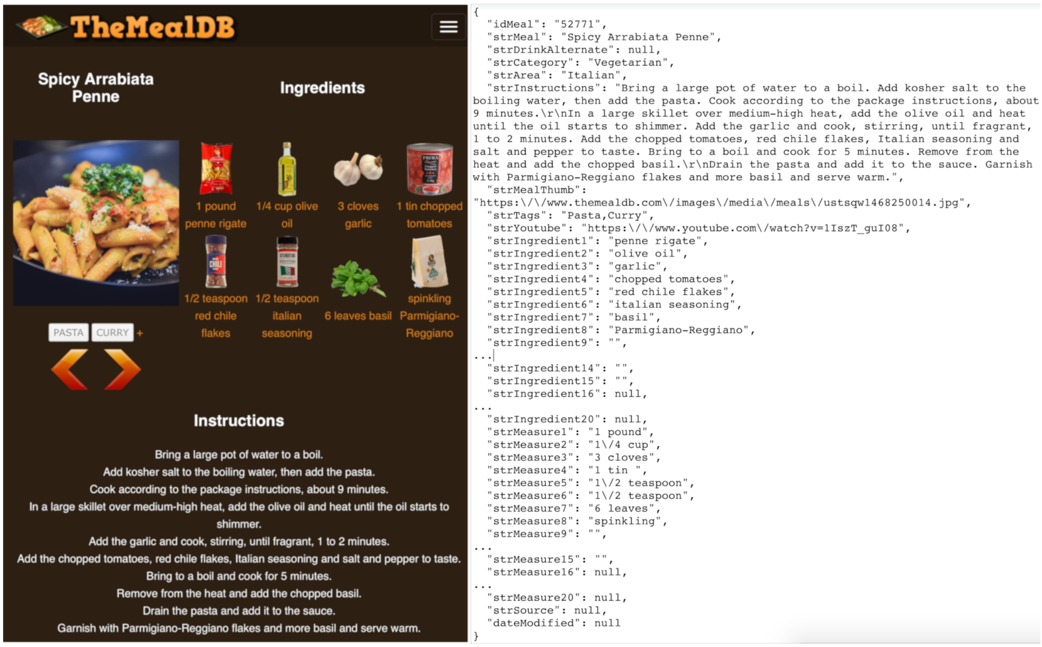


Figure 2.4: theMealDB.com recipe example. Webpage recipe presentation (left) and corresponding feedback from JSON API (right).

bigoven.com This website provides an API with more than 500.000 recipes. The data are available in a JSON dictionary format[6]. The recipes are behind a pay wall[7]. The ingredient list is a list of dictionaries. Each ingredient has a standardized ingredient id. The units are not standardized but can be identified over a unit id which is given. The instructions are available in a text. The text does not refer directly to the ingredients.

yummly.com Yummly.com is a website, that provides more than 2 Million Recipes which are collected from different sources over an API [8]. The requests deliver JSON dictionaries, which include ingredients (without proportions),

[6]http://api2.bigoven.com/web/documentation/recipes
[7]http://api2.bigoven.com/web/documentation/feestructure
[8]https://developer.yummly.com/documentation

nutrition information, and flavors (6 dimensions) [9]. The flavor dimensions are salty, sour, bitter, sweet, piquant, and meaty. They have values between zero and one. The instructions are not provided. These instructions are only available on the original web sources. The API usage is not free, but they offer a limited academic plan (March 2019)[10].

ffts.com and recipesource.com Ffts.com and recipesource.com are websites that collect recipes from different sources. The recipes are downloadable in zipped packages and presented as large text files. The recipes in the text file have a common layout. Simple line breaks, underscores, and spaces construct the layout (see figure 2.5). This layout is not standardized. The recipe content is not annotated.

```
     Title: Catalan Mushrooms With Garlic & Parsley
Categories: spanish
     Yield: 6

     1 lb medium-size white mushrooms,
          - stems; trimmed to 1/2 inch
          quartered
   1/4 c  extra-virgin olive oil
   1/4 c  finely chopped flat-leaf
          -parsley
     2 tb finely chopped fresh garlic
     1    to 2 tsp. coarse salt or sea
          - salt

Put the mushrooms in a large bowl of cold water to soak for 10 min.
Rinse them well and then drain.

Heat a large sautÈ pan with a tight-fitting lid over medium heat. Add
the drained mushrooms to the dry pan, cover immediately, and cook until
all the moisture from the mushrooms is leached out, about 20 min.
You'll know this has happened when you lift the lid for a peek and see the
once-dry pan filled with liquid.

Remove the lid, raise the heat to medium high, and boil until the
liquid evaporates and the mushrooms begin to sizzle in the dry pan but
haven't browned; they'll have shrunk considerably and should be firm when
poked with a fork. Lower the heat to medium and stir in 1 Tbs. of the
olive oil, the parsley, and the garlic. SautÈ, stirring frequently, until
the garlic softens, another 3 to 4 min. Transfer the mushrooms to a
serving bowl, stir in the remaining 3 Tbs. olive oil, and season with salt
to taste (I like to salt them liberally). Serve while hot.


Serve these as a starter or perhaps as an accompaniment to the baked
chicken. Leftovers are great on pizza or added to pasta sauce.
```

```
                         PENA ARRABBIATA SWEENEY

Recipe By     :
Serving Size  : 6    Preparation Time :0:00
Categories    : Sauces

  Amount  Measure       Ingredient -- Preparation Method
--------  ------------  --------------------------------
     1/4    c           Butter
   1        md          Onion -- chopped
   3        lg          Garlic Cloves -- minced
     1/2    ts          Red Pepper Flakes
     1/2    c           Vodka
   1                    Stewed Tomatoes -- or plum
                        -14 oz can, cut up
   6        oz          Heavy Cream
   1        t           Salt
   1        lb          Penne Pasta
   6        oz          Parmesan Cheese -- grated

  Melt butter in a sauce pot over medium heat. Add Onion (diced) and garlic
 (minced) and chopped parsley. Cook to translucent. Add red pepper flakes
 and vodka. Reduce heat and simmer for 2 minutes. Add tomatoes with juice.
 Increase heat to high, while breaking up tomatoes with a spoon. Cook for
 about 5 Minutes.  Stir in heavy cream and salt. Cook until sauce begins to
 thicken (approximately 5 minutes ) Toss with cooked Pena and sprinkle with
 Grated Parmesan Cheese and Serve.
```

Figure 2.5: ffts example recipe (left) and recipesources example recipe(right).

2.4 Digital Structured Food Databases

An ingredient name represents ingredients (food elements) in recipes. For classification grouping and evaluating ingredients of a recipe, further information of all ingredients is important. Important for the used data set is that good coverage of used ingredients from all recipes is given (in the same language, so name matching is easy). The grouping for ingredient classification and reasonable ingredient substitution should be part of the data.

FooDB.ca FooDB.ca is a large comprehensive resource on food constituents, chemistry and biology. It provides information on both macronutrients and

[9]https://developer.yummly.com/documentation/search-recipes-response-sample
[10]https://developer.yummly.com/plans

micronutrients, including many of the constituents that give foods their flavor, color, taste, texture and aroma [11]. FooDB offers the public a freely available food database[9]. The data can be viewed on the website and is also downloadable as a set of files. The central file offers a set of 722 food elements. Each has a food category and a subcategory which describes the food hierarchy. Other files offer matchings from ingredients or food categories to flavors (see table 2.6). Also, very detailed information regarding chemical compounds or nutrition information is available. Often many tables like flavors have NULL entries.

food				flavors	
name	name scientific	food group	food subgroup	name	flavor group
Angelica	Angelica keiskei	Herbs and Spices	Herbs	celery	vegetable
Savoy cabbage	Brassica oleracea var. sabauda	Vegetables	Cabbages	corn	vegetable
Silver linden	Tilia argentea	Herbs and Spices	Herbs	cucumber	vegetable
Kiwi	Actinidia chinensis	Fruits	Tropical fruits	herbaceous	herbaceous
Allium (Onion)	Allium	Vegetables	Onion-family vegetables	sage	herbaceous
Garden onion	Allium cepa	Vegetables	Onion-family vegetables	butter	fatty
Leek	Allium porrum	Vegetables	Onion-family vegetables	cheese	fatty
Garlic	Allium sativum	Herbs and Spices	Herbs	sweet	balsamic
Chives	Allium schoenoprasum	Herbs and Spices	Herbs	vanilla	balsamic
Lemon verbena	Aloysia triphylla	Herbs and Spices	Herbs	lemon	citrus

Figure 2.6: Two example table extracts from FooDB.ca data. Food on four left columns, flavors in right two columns

foodsubs.com Foodsubs.com announces themselves as: "The Cook's Thesaurus is a cooking encyclopedia that covers thousands of ingredients and kitchen tools. Entries include pictures, descriptions, synonyms, pronunciations, and suggested substitutions."[22]. The website is structured hierarchically by food categories. Ingredients are assigned alternative names. The website also names substitution ingredients if some are known. The content appears to be handcrafted with lots of details (see figure 2.7). The website is freely available, but no download of the content as formatted data is given. The underlying HTML code does not follow a standard pattern. Further analysis of the website content is done in chapter 3.2.2. The data acquisition and preprocessing are presented there as well.

openfoodfacts.org The website openfoodfacts.org is a collaborative project with contributors from all around the world. The information is open to everyone. It is a food product orientated website. The product has a set of categories to which it is assigned and a location where this product is available. A table of nutrition information is also available. It is a vast database

[11]http://foodb.ca

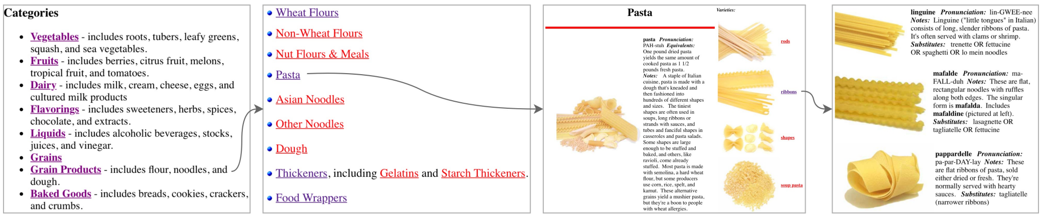


Figure 2.7: foodsubs.com website path to ”Linguine”. Only extract from website content, Central overview of all categories, Grain product categories, All pasta categories (grouped by pasta shapes) with pasta description, Linguine in pasta with shape type ribbon (left to right)

of products for each ingredient. The website provides a search mechanism for food names, bar codes, and other information. Example ingredients have an enormous number of possible products (Olive Oil: 2406 products, Garlic: 2396 products, Linguine: 282 products, March 2019). The number of products is 799877 (March 2019[12])

[12]https://world.openfoodfacts.org

Chapter 3

Machine Readable Recipe Representations

Recipes are documents that describe the creation process of a dish. The underlying semantic structure, that references the ingredients and describes the application of preparation tasks to these ingredients is embedded into the textual instructions. An operation and creation on recipes to gain reasonable new recipes need a computer readable recipe format, that preserves the underlying semantic information of valid recipes. Semantic annotations of the elements of the recipe like the ingredients, their proportions, the applied tasks and the given order preserves the intended structure.

3.1 Data Requirements

Mixing partial instructions of known recipes create new recipes (see section 4.8). A sensible mixing of recipes is done on the instruction level. The recipes need clear instructions. The substitution of ingredients by other similar ingredients reaches further creativity (see section 4.2.5). For this ingredient replacement, the instructions have to name the ingredients clearly. When an ingredient is substitutes by a similar ingredient, this similarity must be provided or calculated. A food hierarchy directly provides ingredients similarity. It is important that the recipe's ingredients have a corresponding element in the food hierarchy. The richer this food hierarchy is, the more ingredient combinations are explorable. The automatic recipe evaluation uses overall recipe characteristics from known recipes, to estimate whether recipes are still valid entities. For the analysis of overall recipe characteristics, the recipes should provide easy accessible semantic data. Newly created recipes have to be presentable in common format. The recipes are presented to many different people for the evaluation so the language should be English. In a Master Thesis frame, it is desirable that the data access is free.

Recipe Subset: Noodle/Pasta Recipes The approach should be able to create arbitrary recipes. Initially the creation process is focused on noodle or pasta recipes [23, 24, 25, 26, 27] to estimate if the creation pipeline performs in a limited domain. This recipe domain limitation can easily be switched to arbitrary recipes (see section 4.2.1). The recipe data needs to include sufficient enough different noodle or pasta recipes. A larger entropy of noodle recipes offers more divers mixing combinations for new recipes.

3.2 Selection of data

The most important requirements for the data selection process are shown in section Data Requirements. Different opportunities for digital recipe data and food hierarchies are available on the internet (see section 2.3).

3.2.1 Selection of Digital Recipe Data Source

The largest recipe data are from Yummly [10] and BigOven [11]. Both provide JSON API. However, both do not fit the necessary criteria: local storage(Yummly) for further analysis or do not offer a fitting free plan for using their API (BigOven). The websites FFTS [19] and RecipeSource [18] are free available but have no clear structure or annotations so huge manual annotation would be needed. Edamam [20] has a free test plan and provides interesting nutritional information for the recipes but does not have optimal annotations like separate data fields for ingredients and proportions. The data set from Recipe recommendation using ingredient networks [3] does not include textual instructions. It only mentions used ingredients without proportions. The website theMealDB [12] offers free access to recipes through a JSON API while allowing using the data for projects like this: "TheMealDB was built in 2016 to provide a free data source API for recipes online in the hope that developers would build applications and cool projects on top of it."[12]. The JSON API for recipes provides recipe information in a JSON dictionary. The language is English. The website theMealDB.com provides around 200 ([12] March 2019) different recipes. An investigation of the number and diversity of noodle recipes shows that a sufficient level is given (see section 3.2.1).

theMealDB.com - Data Analysis The website provides a JSON API to load arbitrary recipes. The recipes are in a dictionary format. Each recipe has 50 key-value pairs, including the title, full-text instructions, Category, Area and Ingredient (chapter 3.2.1). Figure 3.1 shows that the recipes come from all over the world with a focus on western cuisine. The initially limited domain of noodle recipes is sufficiently covered with recipes from different parts of the world (see figure 3.3, right side). In addition, figure 3.3 shows a diversity of these noodle recipes in the category assignment like Pasta, Seafood, Vegetarian or Chicken (see figure 3.3, left side).

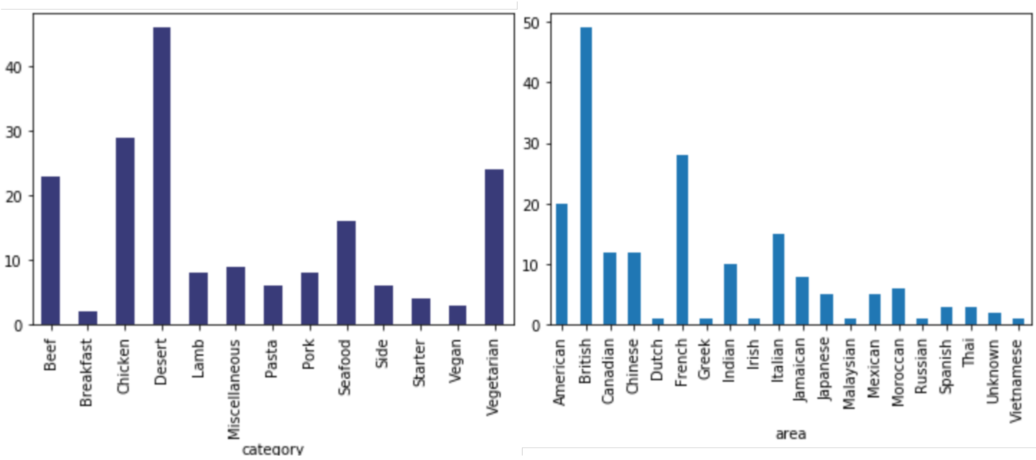


Figure 3.1: Analysis of recipes in theMealDB. Category (left) and area (right)

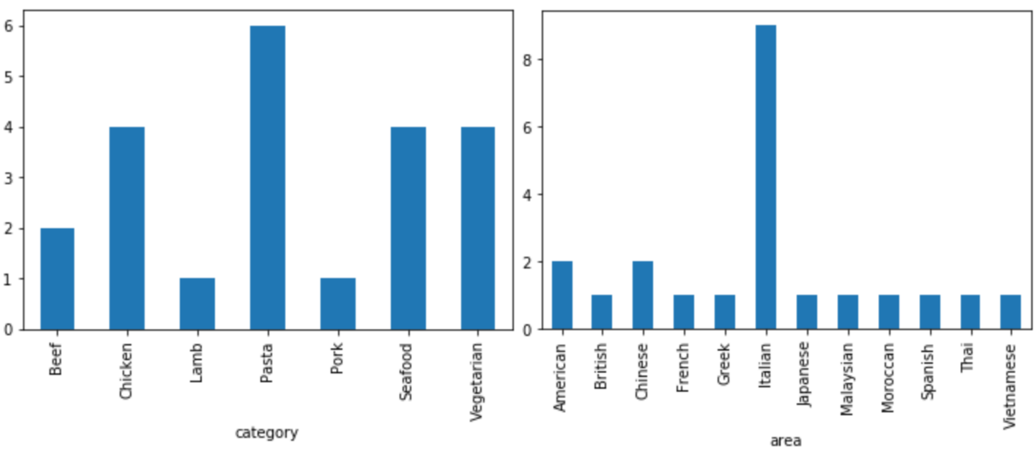


Figure 3.2: Analysis of noodle recipes subset (If in the ingredient list is at least one noodle or pasta ingredient). Category (left) and area/region (right) the recipes are associated to.

theMealDB.com - Dictionary Preprocessing The recipes are downloaded over the JSON API and stored locally so that the algorithm can run without an internet connection. 40 Key Value pairs represent ingredients. Each ingredient has two keys. One for the ingredient name, one for its measure. If a recipe has less than Twenty ingredients, the keys still exist, but the values are empty strings. The Preprocessing changes the ingredient representation to a single key-value pair. This key-value pair consists of the key "ingredients" and an assigned list of dictionaries. Each dictionary in this list represents an ingredient with two key-value pairs. One key for ingredient one key for the measure. An extract from the total set of noodle recipes are presented in figure 3.3. The main difference is that the ingredients are represented as one key for all ingredients. The values are in a null-value free list.

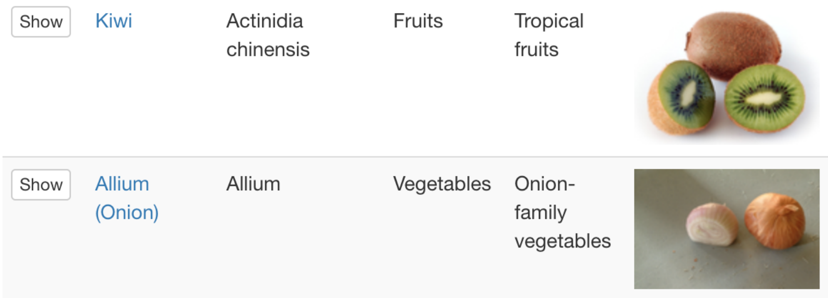
The construction of recipe trees based on the instruction text (chapter 3.3) is done partial computer generated (see section 3.3.2) but mostly manually (see section 3.3.3).

title	area	category	tags	ingredients	instructions
Chicken Alfredo Primavera	Italian	Chicken	[Pasta, Meat, Dairy]	[{'name': 'Butter', 'measure': '2 tablespoons'...	Heat 1 tablespoon of butter and 2 tablespoons ...
Pilchard puttanesca	Italian	Pasta	NaN	[{'name': 'Spaghetti', 'measure': '300g'}, {'n...	Cook the pasta following pack instructions. He...
Spicy Arrabiata Penne	Italian	Vegetarian	[Pasta, Curry]	[{'name': 'penne rigate', 'measure': '1 pound'...	Bring a large pot of water to a boil. Add kosh...
Squash linguine	Italian	Vegetarian	[Pasta, Light]	[{'name': 'Butternut Squash', 'measure': '350g...	Heat oven to 200C/180C fan/gas 6. Put the squa...
Yaki Udon	Japanese	Vegetarian	NaN	[{'name': 'Udon Noodles', 'measure': '250g'}, ...	Boil some water in a large saucepan. Add 250ml...
Laksa King Prawn Noodles	Malaysian	Seafood	[Shellfish, Seafood]	[{'name': 'Olive Oil', 'measure': '1 tbsp'}, {...	Heat the oil in a medium saucepan and add the ...
Pad See Ew	Thai	Chicken	NaN	[{'name': 'rice stick noodles', 'measure': '6o...	Mix Sauce in small bowl.\r\nMince garlic into ...

Figure 3.3: Extract from noodle/pasta recipe table (Pandas Dataframe [1])

3.2.2 Selection of Food Hierarchy Data Source

Different projects provide food hierarchies (see section 2.4) that can be used to infer ingredient similarity (see section 3.1) for the ingredient replacement (see section 4.2.5). The website *foodb.ca* [9] provides a download of tables including 900 ingredients. Each ingredient is assigned to a food group and a subgroup (see figures 3.4, 3.5). An advantage is that for several ingredients, nutrition data are given, which allows estimating health information of recipes. Unfortunately, too few ingredients from recipe database are covered (example not included ingredient names: penne, linguine, noodle, olive oil). To few ingredients means that one of the requirements from section 3.1 is not fulfilled, because the food hierarchy cannot provide a similar ingredient.



Show	Kiwi	Actinidia chinensis	Fruits	Tropical fruits	
Show	Allium (Onion)	Allium	Vegetables	Onion-family vegetables	

Figure 3.4: Example of *foodb.ca* ingredients content. Columns from left to right: name, biological name, food group, food subgroup

	name	name_scientific	description	food_group	food_subgroup
0	Angelica	Angelica keiskei	Angelica is a genus of about 60 species of tal...	Herbs and Spices	Herbs
1	Savoy cabbage	Brassica oleracea var. sabauda	Savoy cabbage (Brassica oleracea convar. capit...	Vegetables	Cabbages
2	Silver linden	Tilia argentea	Tilia tomentosa (Silver Lime in the UK and Sil...	Herbs and Spices	Herbs
3	Kiwi	Actinidia chinensis	The kiwifruit, often shortened to kiwi in many...	Fruits	Tropical fruits
4	Allium (Onion)	Allium	Allium haematochiton is a species of wild onio...	Vegetables	Onion-family vegetables

Figure 3.5: Extract from foodb ingredient table (Pandas Dataframe [1]) showing extracted result from downloadable csv file.

The website *foodsubs.com* [22] offers information about ingredients substitution candidates. The purpose of this website totally fits the idea of replacing ingredient in the cooking process. The website is hierarchical build (see figure 3.6, left). The hierarchy gives a fine granular hierarchical grouping for all ingredients (see figure 3.6, right side ingredient path). Around 6000 ingredient names are listed on this website. Some ingredients names are synonyms so in total around 2000 ingredients are available. The information are on a free accessible webpage. The content is only available as HTML code.

foodsubs.com Data Acquisition and Preprocessing The ingredients are loaded from the website while preserving the path. All elements under "Equipment" are removed because this category only includes hardware tools and no ingredients. Alternative names of the same ingredient are assigned as synonyms in a key-value pair "names" (see figure 3.7). The web-data is embedded in a non-standard HTML code. Every ingredient page has a different formatting structure. An algorithm that uses beatifulsoup[1] handles all exceptional cases. Some ingredients from recipe data set like "beef" are not in the list of ingredients represented by its ingredient name. The website

[1]https://www.crummy.com/software/BeautifulSoup/bs4/doc/

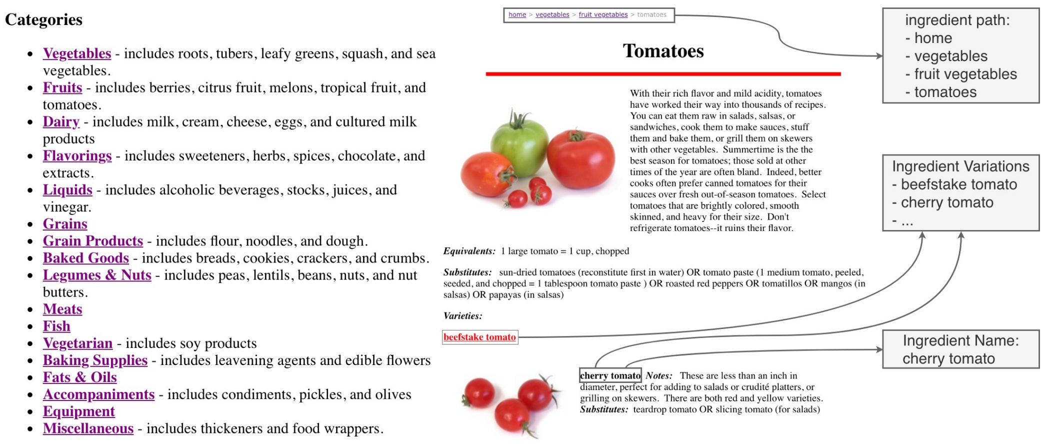


Figure 3.6: Initial page with all categories (left), final website for (tomato) ingredients (right)

foodsubs.com presents some ingredients very fine granular. The ingredient "beef" is the category of ingredients "chuck" or "rib". So categories are also additionally included as ingredients in the resulting dataset. If a category like beef is inserted as an ingredient, the key-value pair type is set to "node" instead of "leaf" (figure 3.7).

name	names	path	type
rialone rice	[rialone rice]	-->food-->Grains-->Rice	leaf
mochi rice	[glutinous rice, sticky rice, sushi rice, swee...	-->food-->Grains-->Rice	leaf
white rice flour	[white rice flour]	-->food-->Baking Supplies-->Non-Wheat Flours	leaf
cream of rice	[cream of rice]	-->food-->Grains-->Rice	leaf

ingredient name

alternative names for ingredient name "mochi rice"

path to white rice flour

Figure 3.7: Extract of resulting table. Some example results if one searches for "rice".

3.3 Genome: Recipe Representation as Tree

The new recipes are created by mixing parts from different instructions. The textual instructions in the recipe describe how to get from the set of ingredients with preparation tasks to the final dish. The order of preparation steps is given by the textual instructions. The ingredient list provides additional information about the precise proportions or the complete ingredient name. Some textual instructions refer the ingredient in a shortened term like: "tomatoes" meaning "250g cherry tomatoes" which is specified in the ingredient list.

3.3.1 Recipe instructions representation inspired by tree-based genetic programming

The instruction text is a linear arrangement of information stored in the sentences. However, instructions in recipes are not always linear procedures. Often separate branches (i.e., cooking in one pot the pasta and parallel the sauce in another pan) are semantically identifiable from the text which allows doing multiple procedure steps in parallel (to save time or allow serving everything the same time). In the end, all recipe components get merged. This approach is inspired by tree-based genetic programming, where nodes are operations and leaves are variables or values. The root node represents the result.

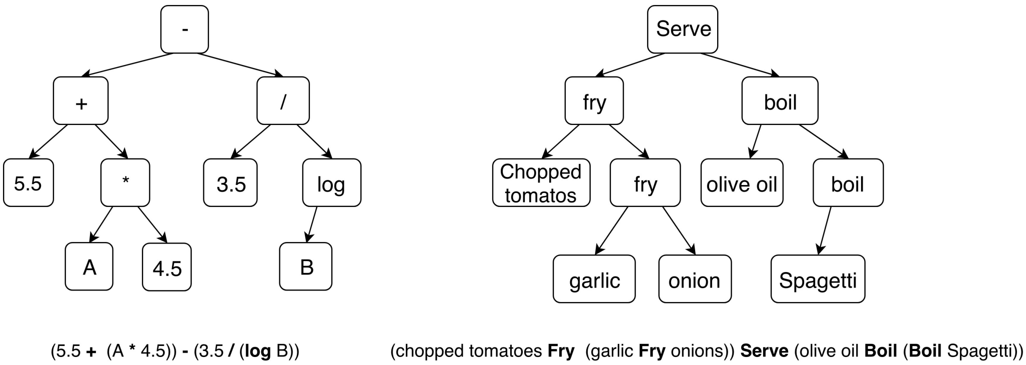


Figure 3.8: Tree-based Genetic Programming Example (left) and example recipe instructions (without proportions) of same tree shape (right).

The fine granular mathematical operations in the nodes correspond to the cooking tasks in the instruction text. These cooking tasks reference temporal results. Figure 3.8 shows how "add chopped tomatoes to fried garlic" corresponds to a partial term of a formula. The variables correspond to the ingredients (ingredient name and proportions). From the tree structure, the ordering of tasks is implicitly given. The right simplified tree in figure 3.8 shows a left branch for the preparation of a sauce and in the right branch the cooking of pasta. In the Evolutionary Algorithm step Recombination, those branches (partial recipes) of different recipes are mixed (see section 4.8) to create novel recipes. Mutation (see section 4.2.5) substitutes the variables (Ingredients) go gain further diversity of created recipes.

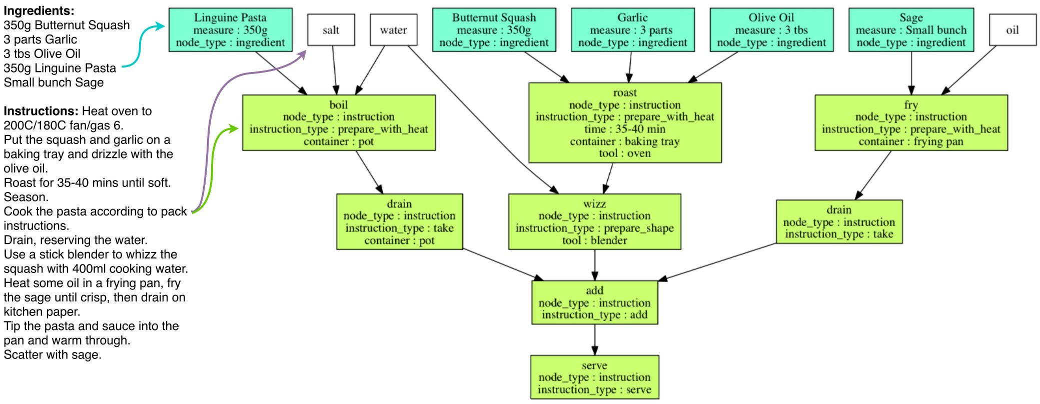


Figure 3.9: Resulting Recipe Tree with annotated data. Extracting ingredient nodes (turquoise arrow), Implicit ingredients (purple arrow) and task node(green). Other examples in Appendix A.1

3.3.2 Ingredient Information - Leaves

The tree leaves represent the recipe's ingredients. Each ingredient has a name and a proportion (see turquise nodes in figure 3.9). These ingredient nodes are constructed by the algorithm based on the preprocessed (see section 3.2.1) theMealDB data field ingredient table. In some Instructions texts some simple ingredients are mentioned which are not represented in the ingredient table like salt or water (these ingredients are assumed to be always available, see figure 3.9). So these ingredients have to be manually extracted additionally from the text. The connection of the ingredient nodes(leaves) are made manually (see section 3.3.3).

3.3.3 Cooking tasks - Annotated Nodes

All other nodes represent an instruction. Children of this instruction node are ingredients or other instruction nodes that represent the result of earlier cooking tasks. The nodes have several meta information which is labeled in this manual preprocessing. Each node has a label *node-type*. The node-type is either instruction-node or an ingredient-node (see section 3.3.2).

Overview of instruction-types The instruction-nodes are grouped into five instruction-type classes. Tables 3.10 and 3.11 shows additional optional key-value pairs for different instruction-types. This annotation makes data analysis possible like which specific instruction types proceeded ingredients. Also which ingredients were used in the same instruction node or subtree.

- **Prepare with heat**: All instructions that include the use of heat. Also, some optional key-value pairs specify the special tool for the heating process or the heat/temperature or time specified for this step (see table 3.10).

Information	Description	Example
Tool	The tool needed for this procedure steps	Oven(incl.mode), microwave, stove
Container	The container in or on the step is processed	Plate, pot, pan
Time Duration	How long heating process should be applied	9 min
Heat	The heat for this procedure step	100 Degree, medium temperature

Figure 3.10: Overview of instruction details: prepare with heat

- **Prepare shape**: Instructions that advising to change the shape or structure of ingredients or subproducts.

Information	Description	Example
Tool	The needed tool for this procedure steps	Knife, blender, rolling pin, injection bag
Container	The container in or on the step is processed	Mixing bowl
Shape	A specific shape or characteristic	Cubes, slices, puree

Figure 3.11: Overview of instruction details: prepare shape

- **Add**: A instruction step which advices to combine ingredients or subproducts without direct further instruction step. In other instructions, it is often an implicit add or combine of ingredients.
- **Take**: There is sometimes the special instruction to take only a part of the partial recipe result. Examples scenarios are to take the cooked noodles from a pot which also includes water or to take meat from a pan with garlic cloves, that were fried together. Often ingredients which are not taken were used to make the procedure step possible (water for boiling noodles) or to add some flavor.
- **Serve**: Always the root node of the recipe tree which defines that all tasks are done.

Chapter 4

Evolutionary Algorithms applied to Recipe Creation

Creating a recipe means the description of a procedure that combines a set of ingredients through preparation tasks. This approach interchanges the ingredients and also the fine granular tasks. These changes yield the creation of novel recipes.

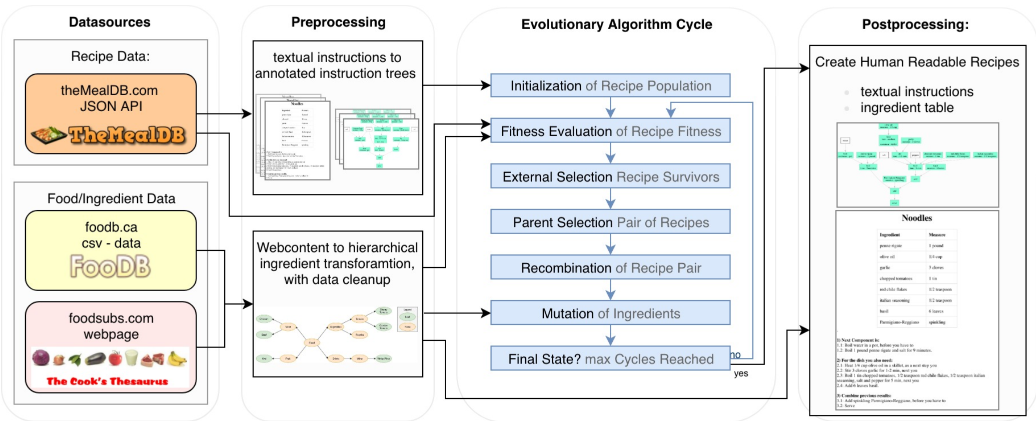


Figure 4.1: Total overview of Recipe Generation Process

This chapter presents the steps in the Evolutionary Algorithm Cycle and the Postprocessing which creates from the tree representations of instructions again human readable recipes. In the graphical overview of the total procedure the components of the Evolutionary Algorithm Cycle (blue, see figure 4.1) are: Initialization of Recipe Population (see section 4.1), Fitness Evaluation of Recipes (see section 4.2.1), External Selection Recipe Survivors (see section 4.2.2), Parent Selection (see section 4.2.3), Recombination of Recipe Pair (see section 4.2.4), Mutation of Ingredients (see section 4.2.5), Final

State? (see section 4.2.7). The transformation of the tree structure into a common human readable recipe format is described in section 4.3.

4.1 Initialization

A set of preprocessed known valid recipes (see figure 4.2 and in Appendix A.1) build the initial population. The recipes have very different instruction-tree shapes and come from various areas of the world (see flags fig 4.2). If the size of the population is larger than the number of prepossessed recipes, each recipe is used mutual times. The algorithm deviates from these recipes through recombination and mutation steps and treats too similar recipes with lower fitness value. The current implementation has a population size of 128 recipes. A parameter can set this size.

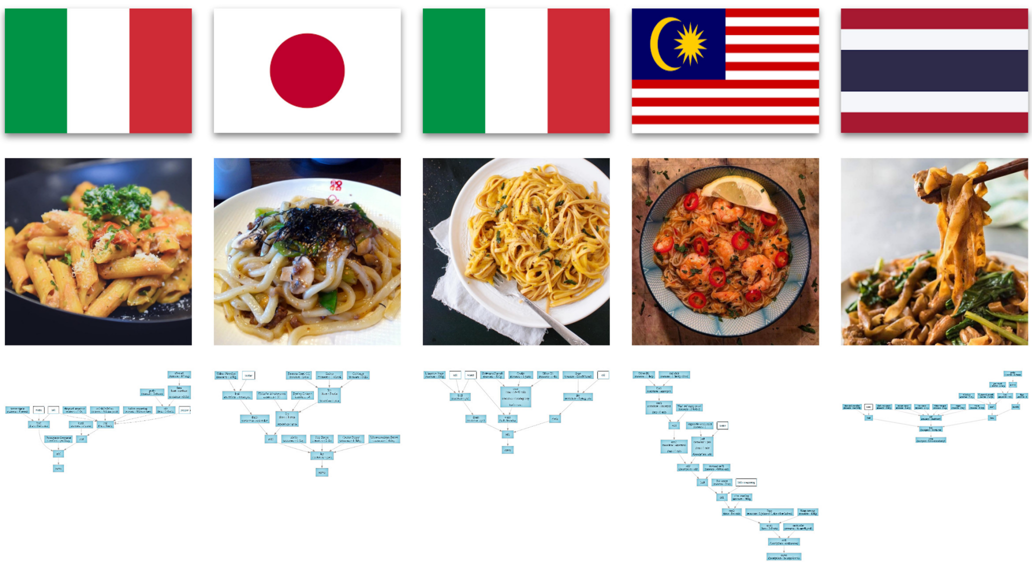


Figure 4.2: Overview recipes for initial population

4.2 Evolutionary Algorithm Cycle

The creative generation of new recipes is done by performing the Evolutionary Algorithm (see figure 4.1). This algorithm generates new recipes in cycles of tasks. The evaluation gives each recipe in the population a fitness value (see section 4.2.1). This fitness value reflects whether this recipe is a valid and novel one. External selection determines dependent on this fitness value, which recipes should be considered for the next cycle and which should be removed because they are not good enough in several dimensions (see section 4.2.2). The number of cycles is a parameter and can be set in the program (see section 4.2.7). Recipes that survived the External Selection are the bases for novel recipes. The recombination merges two random parts of the parent recipes instruction trees (see section 4.2.4). One ingredient of each new merged recipe tree is substituted in mutation (see section 4.2.5).

4.2.1 Evaluation of Recipe Fitness

The Fitness Evaluation performs on the recipe tree a calculation of the fitness value. A weighted sum builds the fitness value of a recipe instruction tree. Multiple fitness criteria are evaluated. Each fitness criteria returns a value between zero and one. The best value is one, and the worst is zero. These values go into a weighted sum where the weights sum up to one. On the one hand, the fitness evaluation calculates whether a recipe is in the area of valid recipes and on the other hand, it evaluates whether a recipe is a novel, meaning a creative, recipe. Novelty is autonomously evaluated based on the similarity of ingredient sets. The recipes validity is evaluated based on five criteria. These criteria are automatically calculated based on the instruction tree (see section 3.8), the recipe database (see section 3.2.1) and the Food Graph (see figure 4.5). More recipe aspects like the price, the availability, and the life cycle assessment of used ingredients need further data. Example extensions of further recipe fitness criteria are outlined in section **??**.

Effort The effort of the recipe preparation procedure is estimated by the number of instruction nodes. All preprocessed recipes are valid recipes. The preprocessed instruction trees have a different number of instruction nodes. The normal distribution of all these valid recipes serves as a rough estimation of how likely a specific number of instruction nodes are. The effort fitness function is based on the normal distribution.

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}, \mu = 11.5, \sigma = 3.95, P_{scaled}(x) = P(x)/P(\mu)$$

The green and black plot in figure 4.3 shows the difference in fitness function (black) and the normal distribution (green). The fitness function is scaled such that the mean has an assigned fitness value of one.

The nodes of type *instruction* from a newly generated recipe tree are counted. This counted number of instruction nodes goes into the fitness function (see green arrow in figure 4.3). The red dot in figure 4.3 shows the resulting assigned effort fitness value for the recipe trees number of instruction nodes.

The fitness value for criteria effort correlates with the likelihood that the new recipe has a valid or reasonable number of instruction nodes. The recombination can enlarge or shorten the recipe in each Evolutionary Algorithm cycle iteration (see section 4.8). This sampling from normal distribution yields that recipes with an uncommon number of instruction nodes will get a low fitness value and will be removed in external selection with a higher chance.

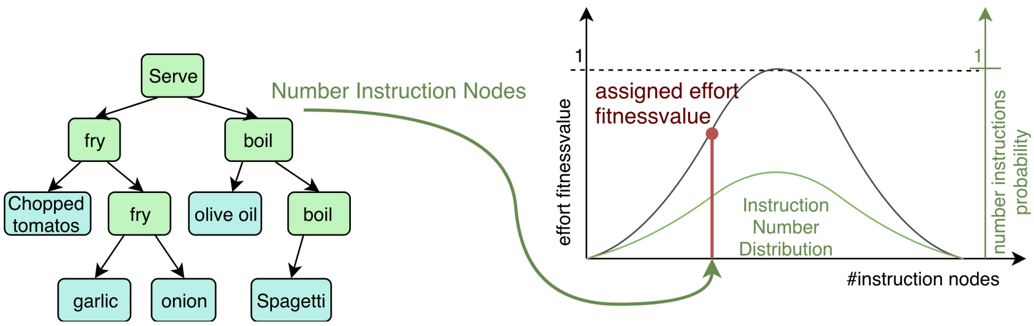


Figure 4.3: Assignment of effort fitness value

Number Ingredients The same method which evaluates the number of instruction nodes compared to the overall distribution of valid recipes (4.2.1) is also applied to ingredient nodes. The number of ingredient nodes in the new generated recipe tree is compared to the normal distribution of known recipes (all recipes in theMealDB 3.2.1 recipe data set, based on the ingredient tables). The assigned *common ingredient number* fitness value results from the scaled normal distribution very realted to the effort fitness value (see figures 4.3, 4.4)

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}, \mu = 8.5, \sigma = 1.8, P_{scaled}(x) = P(x)/P(\mu)$$

This calculation estimates how likely it is that the new recipe has a valid or reasonable number of ingredients. Recombination changes the number of used ingredients. This sampling from normal distribution yields that recipes with a too uncommon number of ingredients will get a low fitness value and will be removed in external selection with a higher chance.

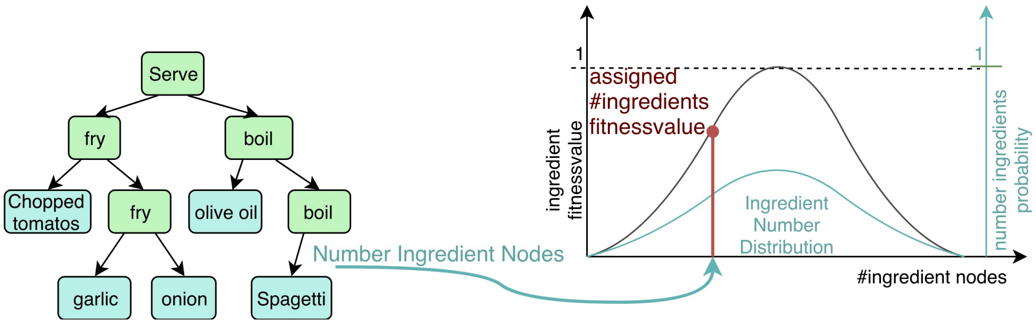


Figure 4.4: Assignment of number ingredients fitness value

Valid Recipe Ingredient Composition This component evaluates if the recipe ingredients cover a pattern of common ingredient categories. The pattern is based on an analysis of know recipes. The first ingredient category contains the main ingredients, which are usually the filling component and can often be recognized because they have the highest share in the ingredient list proportions. Another group consists of the food which is used as side ingredients or toppings. The last group contents all ingredients, like spices and herbs. For each group, a recipe should have at least one ingredient per group to get a return value of one.

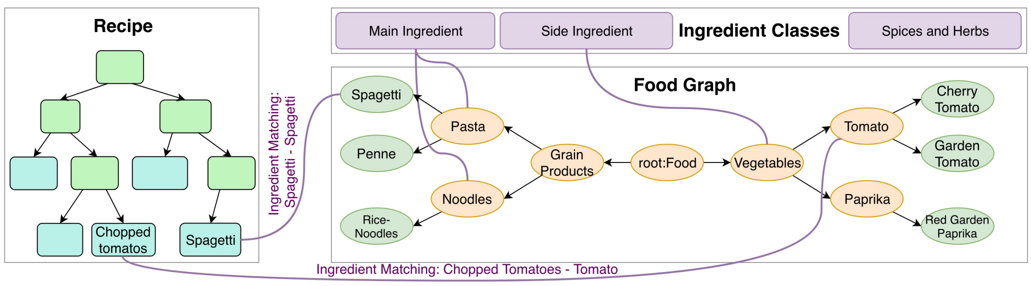


Figure 4.5: Overview of Recipe Ingredient Composition Analysis

The groups are defined by a set of food graph category nodes. The main ingredient group includes, for example, the rice-, noodle-, pasta-, potato-nodes.

Figure 4.5 shows how example ingredients of the recipe tree get an assigned corresponding ingredient node from the food graph (see figure 4.5: Ingredient Matching, purple connections). This assignment is made by a fuzzy string matching (see section 4.2.6). If the path in the food graph from the root node to the assigned node includes at least one node of the specified ingredient group (i.e., main ingredient: pasta, noodle, rice, ...), the recipe composition fulfills this ingredient-group.

Redundant Ingredient Treatment The recombination and mutation can produce a recipe tree that has the same ingredient node multiple times, which should not occur too often. It is treated in the evaluation by a component that divides the number of unique ingredients by the number of total ingredients.

Specified Ingredients It is also possible to specify the necessary ingredients. The current implementation yield is creating new noodle or pasta recipes. So a pasta or noodle ingredient should be present in the recipe tree. Usually, noodle recipes use exactly one noodle ingredient so multiple also different noodle recipes should get a lower score. The calculation return zero if no noodle ingredient is available and one divided by the number of noodle ingredients if the recipe contains noodle ingredients. The ingredient identification is done on the ingredient string bases (see section 4.2.6) and is related to the ingredient composition (see figure 4.5).

Novelty The recipe creation in an Evolutionary Algorithm creates novel recipes. To enforce a deviation from the initial set of recipes, a *novelty* criteria is included in the fitness function. The novelty value is calculated based on the ingredient sets intersection over union (IOU). The most similar ingredient set from the known recipes (kr) and the recipe which is evaluated (nr) results in the highest IOU. The final novelty fitness value results from:

$$f_{NoveltyFitnessValue}(nr) = (1 - max_{kr \in KnownRecipes}(IOU(nr, kr)))^2$$

This term creates high values if the ingredient sets of the evaluated recipe diverge from all known recipes.

4.2.2 External Selection

The external selection depicts recipes that survive the procedure. A parameter defines the ratio of survivors. If more recipes survive, a higher diversity of valid recipes build the set of parents for recombination. A lower survivor ratio yields a faster change and faster deviation from the initial set of non-novel recipes. The current implementation offers n-best, fitness proportional, rank proportional, and tournament selection. This method can be set by a parameter which is in standard set to tournament selection. Tournament selection is the method of choice because it ensures that the best recipe will survive. Also, no recipes are copied compared to rank or fitness proportional external selection. This behavior is wanted because diversity should be high for creative behavior. Also, less fitting recipes get a chance to survive and result in novel recipes by applying recombination and mutation. Tournament selection builds random pairs of two recipes. In each pair, the recipe with the better fitness value survives, the recipe with the lower fitness value is removed (see figure 4.6).

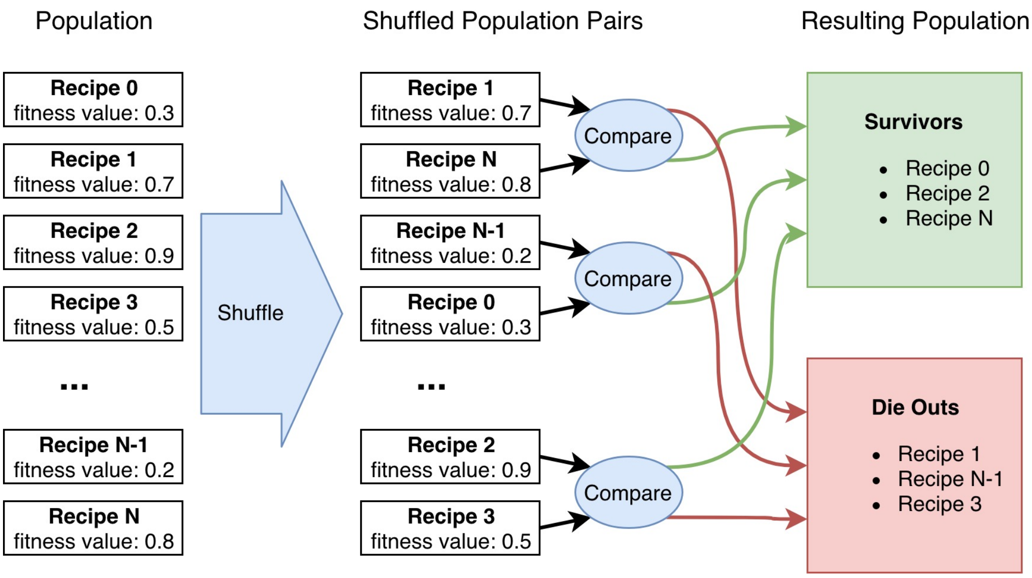


Figure 4.6: Tournament Selection of Recipes

4.2.3 Parent Recipe Selection for Recombination

In recombination, two parent recipes are combined to create a new recipe. These two parent recipes are selected from the set of survivors. In the current implementation, several methods for setting up pairs of parents are implemented. A random assignment depicts by random two recipes without putting back from the set of survivors. Rank and fitness proportional parent selection also pick from the set of survivors. These methods do not ensure that many surviving recipes are used for recombination, so another method is implemented, which ensures that each survivor is equally often used as a parent recipe. This method leads to the highest diversity of parents. The diversity of the parents will result in a higher diversity of created recipes.

4.2.4 Recombination of Instruction Trees

The surviving recipes from external selection build the set of recipes for recombination. The recombination enlarges the population to its original size. For each removed recipe in the external selection step, a new recipe is created from a pair of parents (see figure 4.8). These pairs are two recipes from the survivor set.

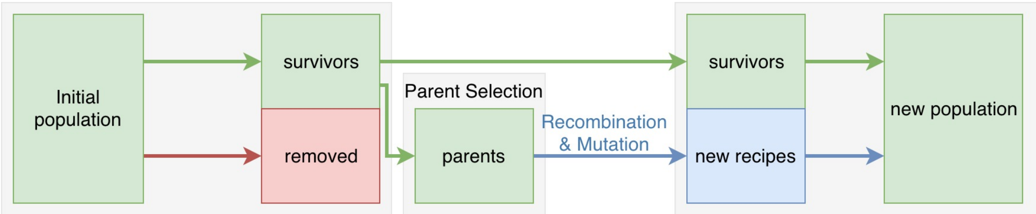


Figure 4.7: Recombination to extend population to restore original population size after External Selection

Main Recipe Tree and Sub Tree Concept In the pair of parents, always one recipe tree is the main tree, and the other tree serves a subtree. Main-tree means that the original parent recipe tree is used, but only one sub-tree is cut off. Sub-tree means that of the second parent recipe tree, only one sub-tree is considered. A sub-tree is a tree where an inner node of the original tree builds the root. From this root on, all recursive nodes and edges build up the sub-tree. The selected subtree of parent 2 replaces the cut of sub-tree of parent one (see figure 4.8).

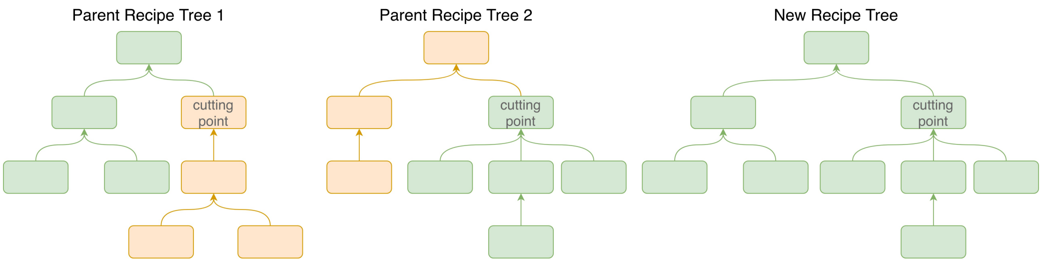


Figure 4.8: Recipe Tree Recombination: Main Tree (left), subtree (center), resulting new recipe tree (right)

Tree Selection The cutting point is selected based on several criteria. The cutting point is never the root node. Also, only a node is selected, which is a branch. So no linear chains of instructions are broken. These chained tasks often belong together and should not be split. If one also allows cutting in these linear chains, less reasonable instructions are constructed. It is also optional to only allow instruction nodes as split nodes. This ensures that each newly created recipe is a truly new set of instructions and not only a mutation of ingredients. This also ensures that the first instruction proceeded to an ingredient is valid and reasonable. It is also implemented, that the subtree of parent 2, which is appended at the cutting point of the main recipe tree (parent 1) should have a similar size than the cut of sub-tree from parent recipe 1. For this method, some subtree candidates from parent two are compared to the cut of sub-tree from parent one, and the most similar sized sub-tree is taken. This selection of a similar sized branch as a replacement yield fewer degenerating recipe tree structures.

4.2.5 Mutation of Ingredients by Food Graph

The mutation changes for each new recipe from recombination one ingredient. This exchange is done by a food database. The original ingredient can be replaced by every other ingredient.

Distance to Probability The probability for each ingredient that it is picked for replacement is different. This probability is based on the similarity of the ingredients. The more similar the ingredients are, the likelier they are to be replaced by each other. This similarity is based on the similarity of the categorization paths of these ingredients. The number of elements in the path which are not the same is interpreted as distance. The shorter the distance, the more similar the ingredients are. The path corresponds to the ingredient classification hierarchy. Food Categories correspond to nodes, and ingredient names correspond to leaves. 3.7.

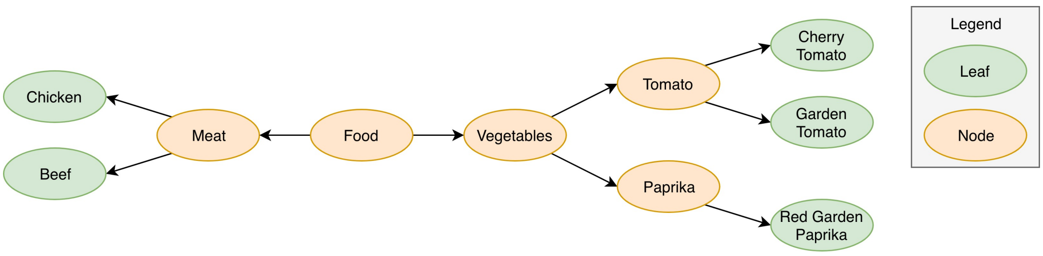


Figure 4.9: Simplified Food Graph Example. Food category nodes are orange, ingredient leaves are green.

If in the original recipe 4.9 Garden tomato o should be mutated all distances to other ingredients are evaluated. These distances are variable for the resulting probabilities for each substitution candidate. A parameter α changes the chance of selecting less similar ingredients. This parameter is between zero and one (exclusive). The higher the parameter is set, the higher the chance that only very similar ingredients are selected. An ingredient i has a group $G(i)$ where all other ingredients with the same distance are included.

The probability p_i for the ingredient i to be selected as a new ingredient in the mutation process is calculated by:

1. $p_i = \frac{\alpha(1-\alpha)^{d(i,o)}}{|G(i)|}$, calculate a probability p_i dependent on the similarity to original ingredient o.
2. $ps = \sum_j p_j$, The sum of all probabilities generated by the formula in the first step
3. $p_i = \frac{p_i}{ps}$ Normalize all probabilities so the total sum of probabilities is one

ingredient	path	distance	probability
Cherry Tomato	Food - Vegetables - Tomato	0	73%
Red Garden Paprika	Food - Vegetables - Tomato	2	18%
Chicken	Food - Meat	3	4.5%
Beef	Food - Meat	3	4.5%

Figure 4.10: Substitution Candidates for Garden Tomato based on example food graph(4.9). Garden Tomato has path: Food - Vegetables - Tomato. Probability calculated with $\alpha = 0.5$.

4.2.6 Ingredient Matching in Data Sets

The Recipe evaluation of the ingredient composition (see section 4.5), the recipe evaluation of specific ingredients (see section 4.2.1) and mutation are performed on a ingredient level. For a recipe entity matching, it is necessary that on the one hand the ingredient is stored in the food graph and that this ingredient is also with slightly different writing identifiable by a matching algorithm. The proposed ingredient in the created recipes is matched against the ingredients of *foodsubs.com* data. This matching is performed on the ingredient name strings. The strings are compared with a fuzzy string matching.

4.2.7 Final State

The final state is reached when a setup number of Evolutionary Algorithm cycles are performed. The longer the algorithm runs, the more recipe combinations have been explored. The final population runs through post-processing (see chapter 4.3). This post-processing creates recipes where all inner dictionary entries are valid and also a human readable recipe from the inner tree structure.

4.3 Post-Processing

In each iteration, the recipe trees evolve. The post-processing that is performed after each evolutionary algorithm cycle ensures that the dictionary, including other data fields like ingredient table, textual instructions, and title make sense. This post-processing is completely autonomous. The algorithm automatically creates a formatted human-readable recipe in HTML 4.3 from the generated recipe trees.

Textual Instruction Generation The procedure of recipe instructions is represented in the tree. The root node represents the final preparation task. If a node has child nodes of type *instruction*, those child nodes have to be done in previous steps. If a node has multiple child nodes of type *instruction*, these branches could be regarded as parallel partial recipe instructions. Each partial recipe starts with a node that has only ingredients as child nodes. The resulting ordering corresponds to the inverted ordering of depth-first search in the recipe tree when only considering nodes of type instruction. For each instruction node a sentence is generated automatically by an *node to sentence* method (figures 4.11, 4.12).

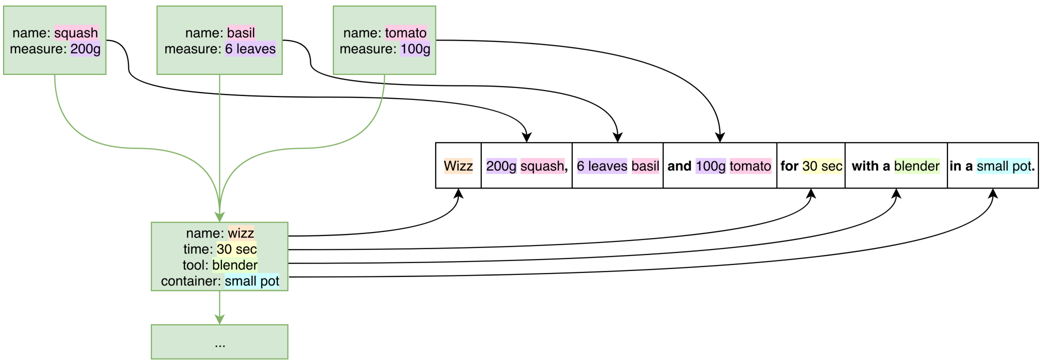


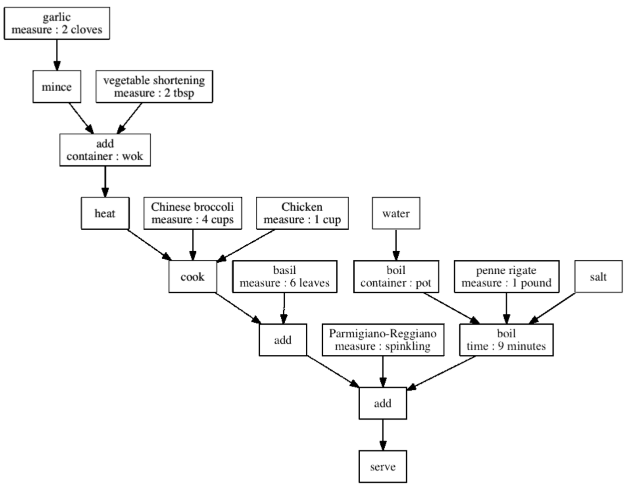
Figure 4.11: From annotated instruction node to human readable sentence

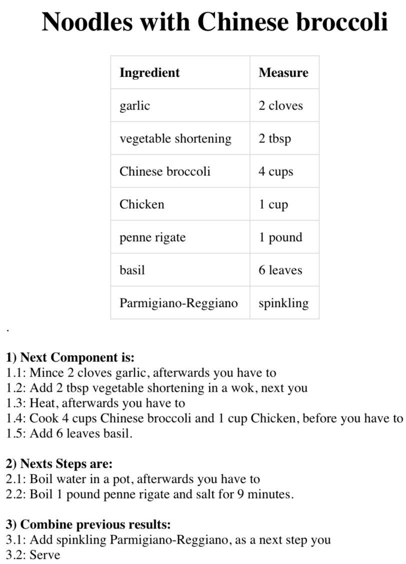
The instruction node **name** is the verb and is assigned as the first word. The first letter is changed to capital. Then all *child nodes* of type *ingredient* are listed. If the number of ingredients is higher than one, the last two are connected with *"and"*. All others are connected by a *","*. If the optional information *time*, *tool* and container are available they are concatenated with fixed connecting words: *"for"*(time), *"with a"*(tool), *"in a"*(container). In the end, only a dot finalizes the generated sentence (figure 4.11). If a node has multiple child nodes of type instruction, a new paragraph is started, and a connecting sentence is inserted, indicating that another part in the recipe starts based on the previous results.

Ingredient Table Update The valid ingredient table can be reconstructed by iterating over all nodes of type ingredient in the recipe tree. The proportions and the ingredient name can be taken from the name and measure value. This is the inverted process of automatic construction of tree leaves from theMealDB recipe database (see section 3.3.2).

Title Generation The recipe title is generated by simple concatenation of the main ingredient and some random side ingredients.

Human Readable HTML Recipe A method is also available that builds an HTML file in a human-readable layout with a recipe title an ingredient table and the textual instructions. The post-processed parts title, instruction table, and instructions are merged.





Noodles with Chinese broccoli

Ingredient	Measure
garlic	2 cloves
vegetable shortening	2 tbsp
Chinese broccoli	4 cups
Chicken	1 cup
penne rigate	1 pound
basil	6 leaves
Parmigiano-Reggiano	spinkling

1) Next Component is:
1.1: Mince 2 cloves garlic, afterwards you have to
1.2: Add 2 tbsp vegetable shortening in a wok, next you
1.3: Heat, afterwards you have to
1.4: Cook 4 cups Chinese broccoli and 1 cup Chicken, before you have to
1.5: Add 6 leaves basil.

2) Nexts Steps are:
2.1: Boil water in a pot, afterwards you have to
2.2: Boil 1 pound penne rigate and salt for 9 minutes.

3) Combine previous results:
3.1: Add spinkling Parmigiano-Reggiano, as a next step you
3.2: Serve

Figure 4.12: Resulting of human-readable html recipe (right) from recipe tree (left) which was generated by the algorithm. Tree nodes only show few information to make the tree better readable

Chapter 5

Evaluation and Discussion

This chapter shows the results of the recipe creation with an Evolutionary Algorithm approach. This section shows a metadata analysis of the recipe creation process and provides the results of a conducted online survey.

5.1 Data Driven Evaluation

The algorithm behavior can be adjusted with a set of Hyper Parameters (see table 5.1). All parameters are chosen to create novel recipes while trying to stay in the valid recipe domain regarding recipe structures. The results of each experiment run are evaluated based on the outcome of intermediate and final created recipes. Also, the overall population is analyzed. For each experiment, all cycles with all intermediate Evolutionary Algorithm results are stored, which allows insights into the recipe creation process.

Hyper Parameter	Description	Example Values
Population Size	Number of Recipes in Population	64, 128
Fitness Evaluation Weights	Importance of fitness relevant criteria	equally distributed
Ratio of Survivors	Percentage of recipe that survive the external selection	0.25, 0.5
External Selection Method	How the external selection performs	tournament selection, n-best
Number of EA Cycles	Food - Meat	5, 10, 20
Mutation Probability Distribution	Rate of probability decay for less similar ingredients	0.5%

Figure 5.1: Table of Evolutionary Algorithm Hyper Parameters

In the experiments, the population size is set to an even number, so external selection methods like tournament selection are more comfortable to perform when one half of the recipes survive. In most experiments, the population has a size of 128 recipes.

Recipe Fitness Function Criteria The recipe validity is evaluated by the automatic fitness function. The assigned fitness values lead to different surviving behavior in external selection. In tournament selection always two recipes are randomly compared. The recipe with better fitness value survives. It is essential that the recipe fitness value corresponds to recipe validity and recipe creativity. The experiments show that the selected criteria yield valid and novel recipes. The criteria which identify recipes validity are based on the recipes normality. The normality is measured by five criteria. The Overview shows that these criteria stay stable close to one. The recipes novelty starts at 0 because the initial population is a set of known recipes and converges to 1 (see fig 5.2).

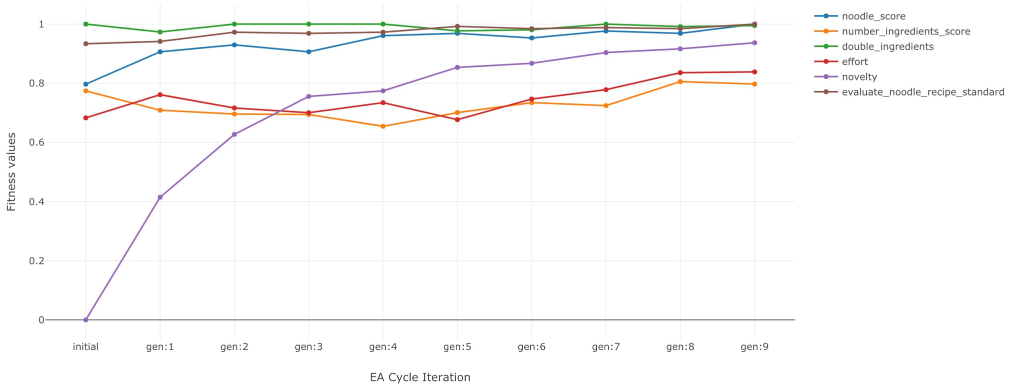


Figure 5.2: Average Fitness Evaluation Criteria over 10 Cycles

The fitness value is automatically calculated. The evaluated criteria go into a function. This function can be changed by different weights of the criteria or making certain criteria mandatory to be fulfilled. Recipe validity is evaluated for example based on the number of used ingredients (see figure 5.3). This number should fit in the overall distribution of all recipes (see section 4.2.1).

Concatenating recipe trees in recombination can lead to large changes that reduce recipes validity. This problem is circumvented by considering the ingredient and instruction amount of each recipe. Also, the Recombination replacement selects a similar subtree from multiple candidates. The number of tested replacement candidates can be adjusted. A smaller number of candidates leads to a higher chance of larger changes. More candidates increase the probability that the tree shape does not diverge much. In the Evaluation this leads to stable numbers for instruction which corresponds to effort and reasonable ingredient list length (see sections 4.2.1, 4.2.1). The validity of novel generated recipes could be improved when the cutting points in

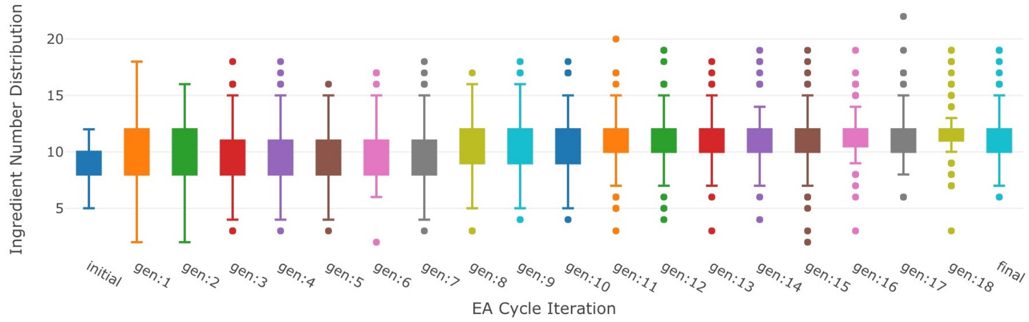


Figure 5.3: Box Plot: number used ingredients distribution in Recipes

recombination are limited. This improved validity is based on the underlining recipe tree structure. An example would be the following up instructions *boil pasta* and then *drain pasta*, which are follow up nodes without branching in the instruction tree. To cut in between does not make sense. The selection of cutting points is adjusted such that it never depicts a cutting point in between a path of only follow up instructions. This restricted cutting points lead to more valid recipes because follow up instructions stay connected.

External Selection Evaluation The rate of survivors is set to 0.5 to preserve the knowledge of the initial recipes or novel good recipes from earlier iterations. However, still, many novel recipes are created and explored. In theory, it could happen that in the final generation, still known recipes are present. In the fitness evaluation novel, recipes get a better fitness value. Figure 5.4 shows that with the presented Hyper-Parameter setup, the last elements from the initial population of recipes die out after around four generations.

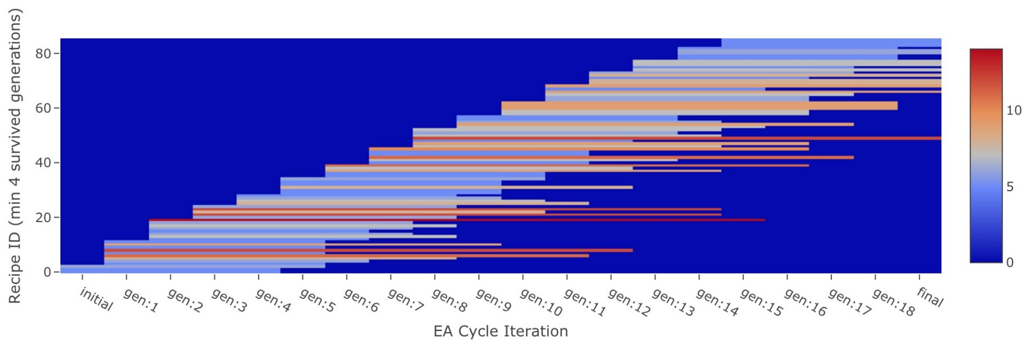


Figure 5.4: Survivor History Overview

In this figure, only recipes with a lifetime of at least four generations are plotted. The color corresponds to the number of survived generations. Figure 5.4 shows that the initial population members die out at least in the intermediate single digit cycle. With a lower survivor rate and higher importance of novelty, the initial population dies out much faster.

The External Selection is implemented with an opportunity in choice of method (n-best, fitness-proportional, rank-proportional, and tournament selection). Most runs are set up to tournament selection because it also allows bad candidates to survive and evolve in further iterations. This increases the chance of higher diversity and so creativity.

The number of EA cycles is set in the lower double digits. With a higher number of cycles, more recipes are explored, and convergence to the desired fitness criteria is reached. The experiments show that in most cases after ten iterations, convergence is reached (see fig. 5.5). Also, the distribution of fitness values shrinks. The higher the iteration, the more the recipes differ from the initial set of valid recipes. So more iterations are only reasonable if the fitness function can classify recipes validity.

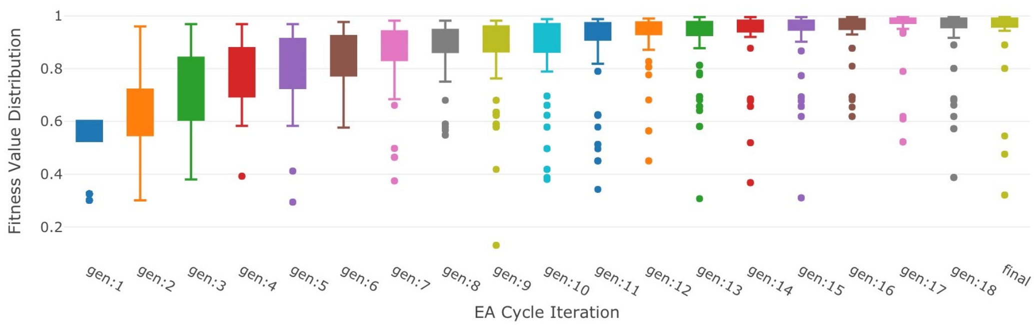


Figure 5.5: Box Plots for Recipe Fitness over Cycles

The Mutation Probability Distribution changes the probability of every ingredient to be picked in the mutation regarding its similarity. This value lies between zero and one and defines how the decay of probability with increasing distance in food graph distance. A higher number yields more stable recipes because the proportional chance for large changes in ingredients is lower but to explore the whole food graph needs more cycles. Smaller changes in the food category are necessary when the connected preparation step is not changed.

However, still, the mutation leads to many explored ingredients over cycles (see fig. 5.6). The slightly decreasing rate of novel explored ingredients is based on the fact that some ingredients of the local food graph were explored before.

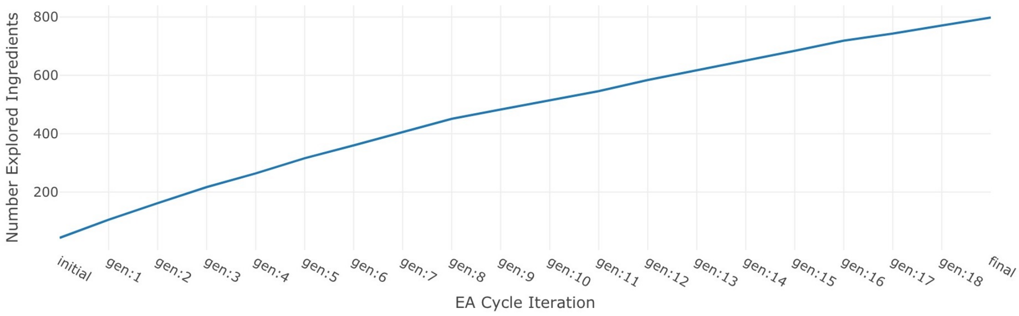


Figure 5.6: Number Explored ingredients over Cycles

Ingredient Matching Evaluation The Recipe creation approach creates new recipes based on known recipes from a recipe database. The mutation of ingredients and the evaluation of the used ingredients is performed by the use of the hierarchical structure of the food graph (see sections 4.2.1, 4.2.1, 4.2.5). The matching is performed based on the ingredient name strings (see section 4.2.6). Figure 5.7 shows the matching of the recipe ingredients against the foodsubs ingredients. For a recipe entity matching, it is necessary that on the one hand, the ingredients are stored in the food graph and that this ingredient is also with slightly different writing identifiable by a matching algorithm.

1	theMealDB	foodsubs	matching
2	Eggs	Eggs	match
3	Baking Powder	baking powder	match
4	Vanilla Extract	vanilla extract	match
5	Oil	oil	match
6	Pecan Nuts	Nuts	subclass
7	Raspberries	raspberry	match
8	Beef Brisket	Beef	subclass
9	Salt	Salt	match
10	Onion	onions	match
11	Garlic	Garlic	match
12	Thyme	thyme	match
13	Rosemary	rosemary	match
14	Bay Leaves	Indonesian bay leaves	subclass
15	beef stock	Beef	subclass
16	Carrots	carrot	match
17	Mustard	preserved Sichuan musta	subclass
18	fajita seasoning	Italian seasoning	no match

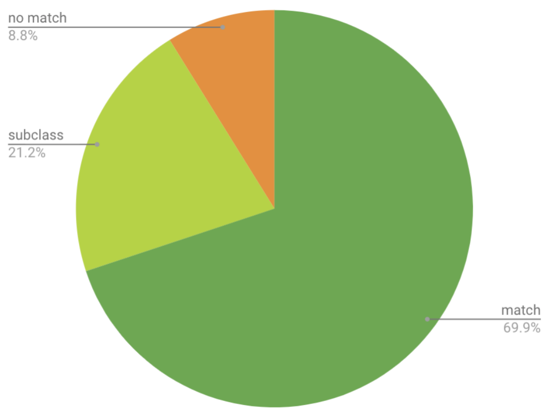


Figure 5.7: Matching Evaluation Overview

This algorithm was evaluated based on more than a hundred matches. It turns out that the database of foodsubs.com [22] outperforms foodb.ca [9] in number of ingredient names (foodb.ca ~900 and foodsubs.com ~6000) and in overlapping with ingredients. The foodsubs evaluation shows coverage of 91.2%. This coverage is based on 69.9% perfect matches and 21.2% of subclass coverage. Subclass means that the match provides two different levels of the food name such that one is the subclass of the other ingredient like *chicken* and *chicken breast*. No match was considered for example when *fajita seasoning* is matched to *Italian seasoning* (see fig 5.7. Also, when this matching is no total failure, it is considered as no match (see the last row in figure 5.7).

5.2 Resulting Recipe Evaluation - Survey

Multiple runs of the Evolutionary approach created thousands of recipes. These recipes have a fitness value calculated by the autonomous fitness function. Only humans can deliver reliable feedback if recipes are understandable. Humans can also estimate whether a recipe will be edible and tasty. An online survey gains real human feedback for the created recipes.

Survey setup The survey is shared as a Google Form over social media channels. The layout is easily accessible on any device. The recipes are presented in the post-processed format (see section 4.3).

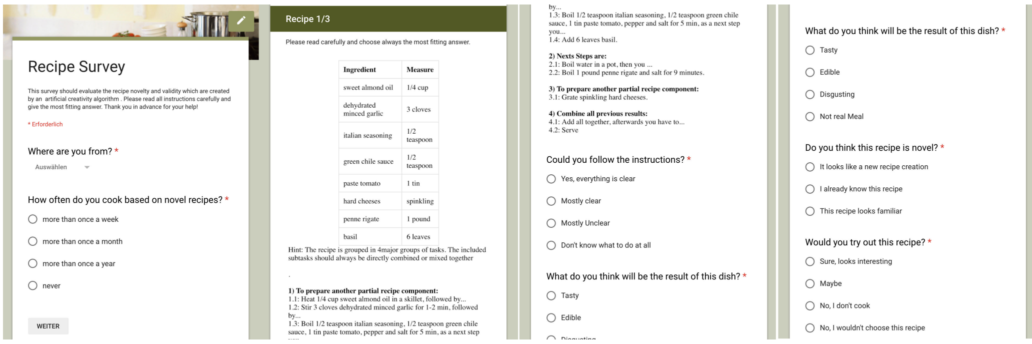


Figure 5.8: The Google Form Survey (side by side)

Fifteen recipes are presented. The presented recipes are from three groups. The initial population is represented by five recipes (see appendix 3.3). The remaining ten recipes are created recipes, where 8 of them are from five different runs of the algorithm after ten evolving cycles (final evolved population) and two recipes from the algorithm after one cycle (early evolved population). The recipes are selected by random choice of recipes within the three highest fitness values of the population. The selection of a recipe is resampled, if it is too similar to the previously selected recipes, to gain a bigger derivation of presented recipes. A participant has to evaluate at least one group of three recipes. The groups always include at least one final

- Initial Questions
  - (Q1) Where are you from? (A) Africa; Asia; Australia; Europa; North America; South America
  - (Q2) How often do you cook based on novel recipes? (A) more than once a week; more than once a month; more than once a year; never
- Recipe Questions
  - (Q3) Could you follow the instructions? (A) Yes, everything is clear; Mostly clear; Mostly Unclear; Don't know what to do at all.
  - (Q4) What do you think will be the result of this dish? (A) Tasty; Edible; Disgusting; Not real Meal
  - (Q5) Do you think this recipe is novel? (A) This recipe looks familiar; It looks like a new recipe creation; I already know this recipe
  - (Q6) Would you try out this recipe? (A) No, I wouldn't choose this recipe; Maybe; No, I don't cook; Sure, looks interesting

Figure 5.9: Recipe Survey Question, Answer Pairs

recipe. The recipe group assignment is done randomly. Each participant has to answer two initial questions. The first question asks for the participant's origin (continent), and the second asks for its cooking behavior regarding how often the participant cooks based on novel recipes. For each recipe four questions are presented (see Overview Fig 5.9). In the end, each participant could add an optional overall comment.

Survey Evaluation The survey got feedback by more than 50 participants. Each participant evaluated 3 or 6 recipes. This lead to the fact that each of the 15 recipes was evaluated by at least 10 and most 15 participants. The participant's origin was asked in the beginning (see fig: 5.9 Q1). Almost everyone was from Europe. The cooking behavior was evaluated in question 2 (see fig: 5.9 Q2). The majority cooks more than once a month based on novel recipes (see fig: 5.10). Another third cooks more than once a year based on novel recipes.

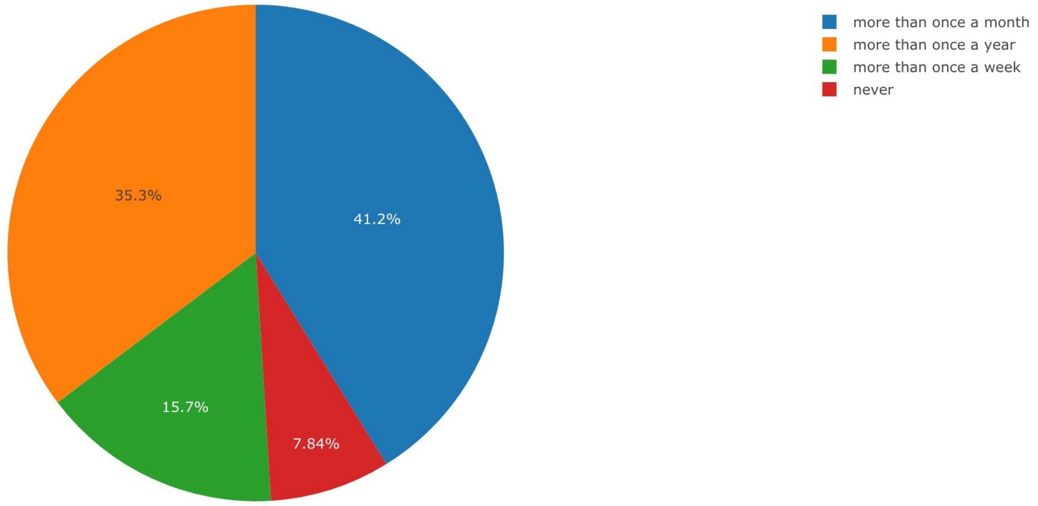


Figure 5.10: Survey Evaluation: How often do you cook based on novel recipes?

Recipe Validity Evaluation The recipe Validity was measured with two questions. The first was, whether the instructions are followable (see fig: 5.9 Q3). The second validity evaluating question asks what result the participant expects (see fig: 5.9 Q4).

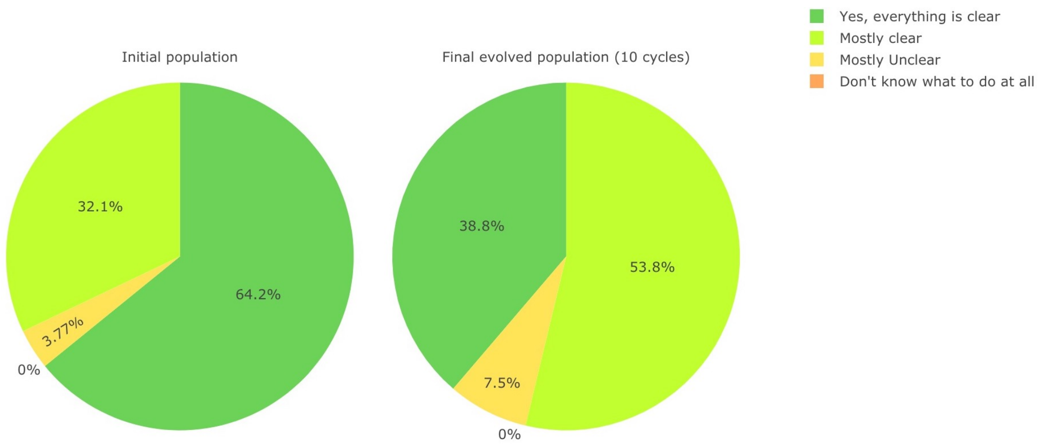


Figure 5.11: Survey Evaluation (Q3): Could you follow the instructions?

The answers to the first question (Q3) show that the initial recipes have a higher rate if all instructions are perfectly clear. The combined distribution if the recipes are at least mostly clear. In this comparison, both groups are close together. The lousy feedback category ”don’t know what to do at all” was never selected. The ”mostly unclear” answer option was in both cases at a shallow rate (see fig: 5.11).

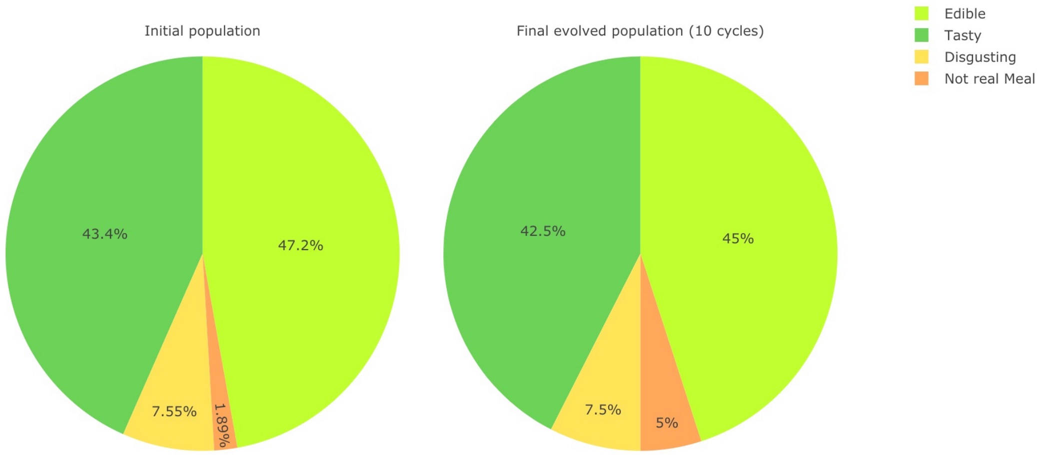


Figure 5.12: Survey Evaluation (Q4): What do you think will be the result of this dish?

The second validity evaluating question (Q4) was "What do you think will be the result of this dish?". No significant difference is visible. In both classes (initial valid known recipes and novel created recipes) almost on half is judged to be tasty and almost the other half is judged to be edible. The remaining parts are with 7.5% "Disgusting" in both classes and "not real meal". Only the classification "not real meal" is slightly higher from 1.89% to 5% in the evolved recipes. But in general both recipes are on same level (see fig: 5.12).

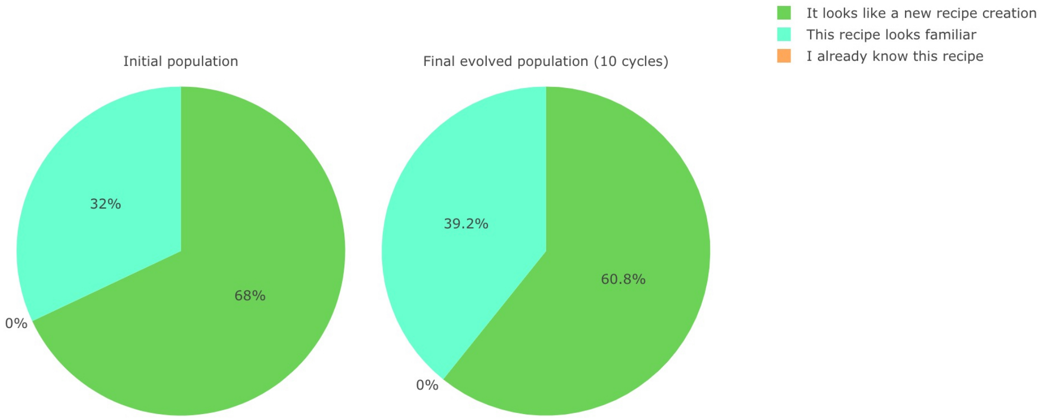


Figure 5.13: Survey Evaluation (Q5): Do you think this recipe is novel?

A third recipe evaluating question (Q5) asks if the participant thinks that this recipe is novel or if the participant knows this recipe or it looks familiar. The participant's answers show that nobody knows the recipes. In both classes on third of the recipes are voted to look familiar, and two third are judged to look novel (see fig: 5.13).

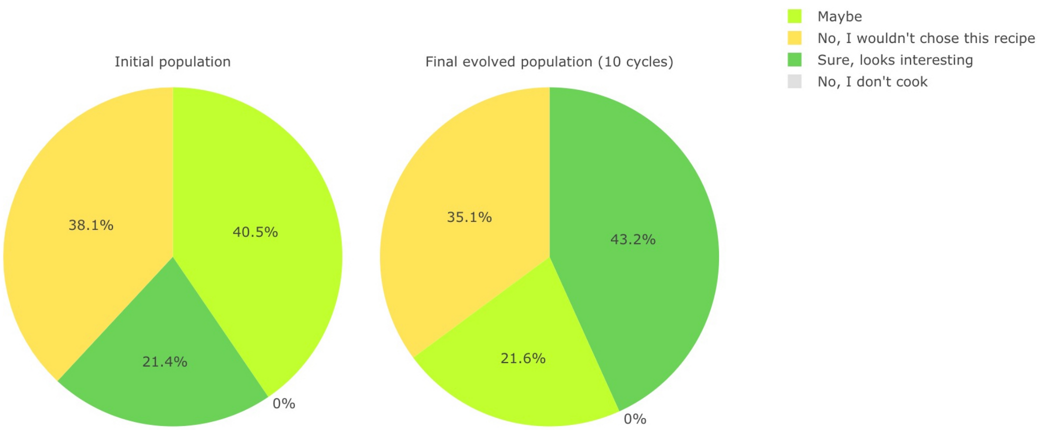


Figure 5.14: Survey Evaluation (Q6): Would you try out this recipe?

A final question (Q6) asks whether the participants would try out these recipes. The novel created recipes outperforms the initial recipes (see fig: 5.14). For both classes, it is interesting that the participants mentioned in a final optional comment field that when they answered that they "would't choose this recipe", it could be based on two reasons. One would be that they don't think that this will be an excellent satisfying dish in a general sense, the other reason which was named (in a final commentary field), is that the participant is a vegetarian, which thinks, that in general, this might be a good recipe but it fits not their diet, allergies or veggie constraints.

5.3 Discussion of Issues

The algorithm stores for each run for each cycle all intermediate steps of the population. All changes could be tracked, and also intermediate recipes are visualized in tree and text layout.

From an analysis of the recipe trees, two problem cases occur. One group is based on the limited automatic fitness function. The second group is based on the wrong ingredient matching. Some recipes are not preparable because the connected preparation node does not fit to the underlying ingredients. This could happen because of three reasons. If substantial changes in mutation yield to totally different ingredient types, the mutated ingredient does not fit the connected preparation step anymore. If in recombination a whole different kind of subdish (tree of instructions) is merged this could be not sensible. It could also happen that in mutation the named ingredient is misidentified by its name, such that the mutation operation assigns a similar ingredient from the wrong identified food category which makes no sense regarding connected preparation mode again. These scenarios could be solved by a fitness function that is sensitive to ingredient preparation mappings. A better ingredient entity recognition and matching also circumvent one type of failure.

Chapter 6

Conclusions and Future Work

This Master Thesis presents a new approach that successfully creates novel recipes. The conclusion outlines the contributions of this work. Feature Work presents various ideas for an extension of this approach.

6.1 Conclusion

The novel recipes are created in an Evolutionary Algorithm inspired by genetic programming. High numbers of Evolutionary Cycles and population size are only possible because all steps of the Evolutionary Algorithm run fully autonomously and creates mostly valid and novel recipes. The Recombination of recipes on the instruction tree basis preserves semantic dependencies. The use of food hierarchies in the Mutation allows arbitrary ingredient combinations while exploring ingredient substitutions sensibly regarding their food type. The Autonomous Fitness Evaluation presents an initial set of recipe criteria that can be evaluated autonomously without human feedback to steer the creation process in the desired domain of novel and valid recipes. It is shown how many recipe fitness dimensions can automatically be calculated. The extensibility of this autonomous fitness evaluation is presented in Future Work. The annotations of the instruction trees and their implicit order make it possible to autonomously create in a postprocessing human readable recipe representations with a common layout. Very positive feedback from an online survey shows that this approach creates recipes that are accepted by humans.

6.2 Future Work

The recipe generation shows promising results. The Algorithm is designed to allow several additional optimization and further features. Many of these features need special data that stores further information about ingredients and cooking.

6.2.1 Enlarge Fitness Evaluation Dimensions

The most difficult part is the estimation of whether a recipe is good or not. The term good regarding recipes has different dimensions. Some of them can be calculated or estimated to produce an evaluation with broader scope regarding quality

Ingredient Fitting - Taste The taste of the resulting dish cannot be calculated. Scientific work[3, 5] estimates recipe preferences and fitting of ingredients that could be used as an estimate for the resulting taste.

Regional food characteristics The recipe evaluation or the ingredient mutation can be tuned to local/regional favors. Different cultures have, for example, sometimes different preferences of how spicy a dish should be or how they compose flavors in their recipes [28].

Health Information The healthiness of a recipe is based on multiple facets. A weighted sum over the ingredients nutrition information can give a good estimation to the total included nutrition (preparation methods can change them). This nutrition distribution can be compared to personal needs, including allergens and diets. Data sets like foodb.ca [9] could provide those data.

Sustainability Sustainability should also be considered when a recipe is created. On the one hand we have the energy consumption of the preparation and on the other hand all the used components. Components are the tools we need and the ingredients we use. Tools and ingredients have to be produced and often transported. These aspects can be measured and evaluated in dimensions like water-footprint or CO2 footprint (Life Cycle Assessment). Tools need be produced usually once for multiple preparations of recipes, but sometimes tool substitutions are available as well. Same scenario to ingredients. In many countries, we have access to the ingredients of the whole world and not only regional and seasonal ingredients. Not always does the footprint correlate with the price. I suggest to include those sustainability costs as well.

Price The price is dependent on the local supply. However, if all ingredients are available, the price of a dish can be estimated by summing up the ingredients multiplied by their local prices. On also can consider energy costs or tool costs and possible waste of not perfect fitting proportions of bought ingredients.

6.2.2 Mutation Optimization

The Mutation step covers the substitution of ingredients. This substitution of ingredients has dependencies on other recipe information. The treatment of these dependencies can be optimized.

Preparation Method Mutation When ingredients are mutated, the connected instruction nodes should be mutated as well such that the instruction is doable. Not every ingredient combination can be cooked or fried. The fact if a set of ingredients can be boiled depends on aspects like how liquid the used ingredients are. The annotation of the preparation nodes should be used for reasonable replacements. A "prepare with heat" node should be replaced with another "prepare with heat" node. The heating method has to be assigned. Knowledge of what is possible and what is not can be derived from recipe instruction trees. An analysis of which ingredients or ingredient groups were processed with specific preparation methods shows can be included in the fitness function, the recombination constraints, and mutation constraints.

Available Products The mutation operator could be modified that only ingredients which are available in the local store are used. This can be implemented in the Food Graph.

Ingredient Substitution Probabilities The probabilities for an ingredient to be the substitution for another ingredient is based on its food graph distance. This probability could also be based on a substitution network [3]. Another alternative could be to follow the substitute suggestions from websites like foodsubs.com[22] (see substitutes in figure 3.6).

Number of mutated ingredients The current Approach mutates in each recipe exactly one ingredient after Recombination. This number of ingredients to be mutated could be sampled by a decreasing probability for higher numbers.

6.2.3 Food and Recipe Database Generation

The recipe data set is gathered by an automated usage of the JSON API by theMealDB.com and partial automated preprocessing. It would be great if the instruction tree generation could be done entirely automated instead of partial automated. The used ingredients have no standardized proportions. This standardization would also be essential to evaluate, for example, nutrition information.

Automatic Tree Structure Generation from Instructions The last step, which makes this approach not fully autonomous is that the recipe instruction trees have to be constructed partially manual. This autonomous construction will be very challenging. An NLP approach has to gain the semantic information out of the instruction text. With entity recognition, the ingredients in the instruction text have to be matched against the ingredient table. From POS tagging, the verb can be used as an instruction node. This verb has to be classified to be annotated in the given structure (heat or shape instruction, see section 3.3.3) One of many problems is that often intermediate steps do refer the original by substitute names for the resulting partial recipe components. Chopped tomatoes, fried onions, and seasonings are named sauce. This matching is tough. Far more aspects have to be considered.

Proportion Standardization The proportions should be mutated as well, which is not done in the current approach. The Food Graph could be extended to hold also normal distributions of proportions for each ingredient derived from analyzed recipes. If no recipe uses this ingredient an approximation could be sampling from the most similar next ingredient.

6.2.4 Applications Scenarios

The capabilities of this approach could be used in different food-related domains. The transfer of recipes to the local cuisine preferences could be applied by giving the fitness value importance of product availability and local taste. Besides, the mutation operator could be changed s.t. only available local products are selected. Another application could be to transfer a recipe to personal diet restrictions. The underlying ingredients are evaluated against the personal needs so proportions and selection of ingredients could be optimized for a personal fit. The same would be doable for reducing price, effort, or Life Cycle Assessment[29].

Appendix A

Appendix

In the appendix figures of the initial recipes and corresponding recipe trees are shown.

A.1 theMealDB Recipes - Tree Represena-tion

This section outlines some recipes [23, 24, 25, 26, 27] which are imported from theMealDB [1] and are transformed to a tree structure. The visualization are done with graphviz[2]. The node colors in the figures for the initial recipe trees are: white for implicit ingredients, turquoise for ingredients, listed in the ingredient table and green for instructions.

Spicy Arrabiata Penne The ingredient list is: *1 pound penne rigate, 1/4 cup olive oil, 3 cloves garlic, 1 tin chopped tomatoes, 1/2 teaspoon red chile flakes, 1/2 teaspoon italian seasoning, 6 leaves basil, spinkling Parmigiano-Reggiano* The instructions of Spicy Arrabiata Penne are: "*Bring a large pot of water to a boil. Add kosher salt to the boiling water, then add the pasta. Cook according to the package instructions, about 9 minutes. In a large skillet over medium-high heat, add the olive oil and heat until the oil starts to shimmer. Add the garlic and cook, stirring, until fragrant, 1 to 2 minutes. Add the chopped tomatoes, red chile flakes, Italian seasoning and salt and pepper to taste. Bring to a boil and cook for 5 minutes. Remove from the heat and add the chopped basil. Drain the pasta and add it to the*

[1] theMealDB.com

[2] https://graphviz.readthedocs.io/en/stable/manual.html

sauce. Garnish with Parmigiano-Reggiano flakes and more basil and serve warm."[27] The resulting recipe instruction tree is presented in figure A.1.

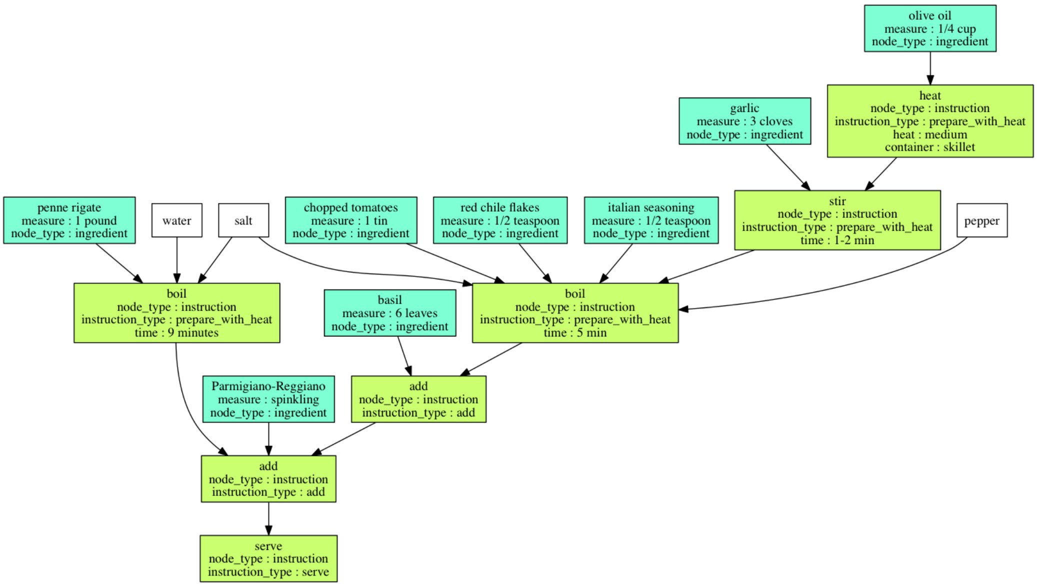


Figure A.1: Resulting Recipe Instruction Tree: Spicy Arrabiata Penne

Yaki Udon The ingredient list is: *250g Udon Noodles, 2 tbs Sesame Seed Oil, 1 sliced Onion, 0.25 Cabbage, 10 Shiitake Mushrooms, 4 Spring Onions, 4 tbsp Mirin, 2 tbs Soy Sauce, 1 tblsp Caster Sugar, 1 tblsp Worcestershire Sauce* The instructions of Yaki Udon are: "*Boil some water in a large saucepan. Add 250ml cold water and the udon noodles. (As they are so thick, adding cold water helps them to cook a little bit slower so the middle cooks through). If using frozen or fresh noodles, cook for 2 mins or until al dente; dried will take longer, about 5-6 mins. Drain and leave in the colander. Heat 1 tbsp of the oil, add the onion and cabbage and sauté for 5 mins until softened. Add the mushrooms and some spring onions, and sauté for 1 more min. Pour in the remaining sesame oil and the noodles. If using cold noodles, let them heat through before adding the ingredients for the sauce – otherwise tip in straight away and keep stir-frying until sticky and piping hot. Sprinkle with the remaining spring onions.*"[26] The resulting recipe instruction tree is presented in figure A.2.

Squash Linguine The ingredient list is: *350g Butternut Squash, 3 parts Garlic, 3 tbs Olive Oil, 350g Linguine Pasta, Small bunch Sage* The instructions of Squash Linguine are: "*Heat oven to 200C/180C fan/gas 6. Put the*

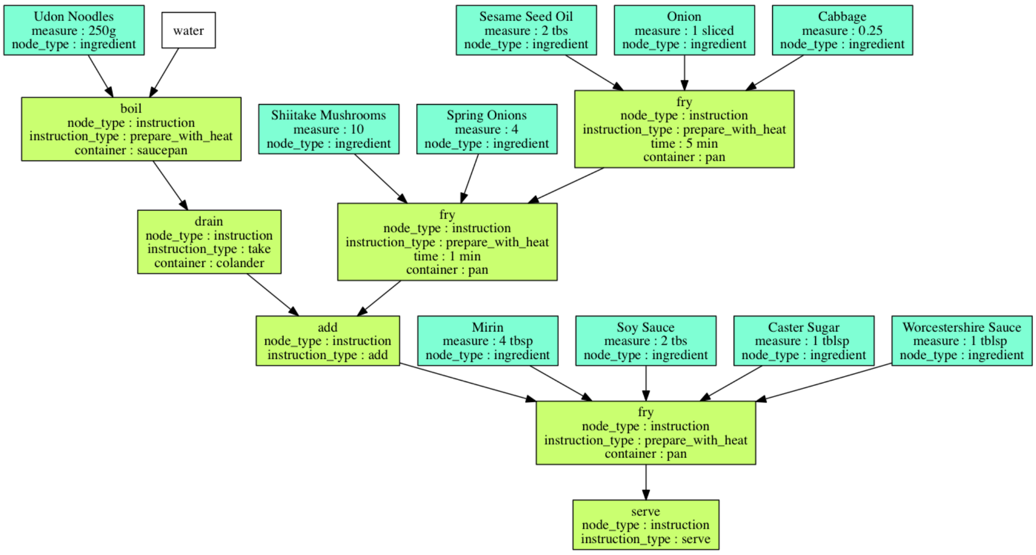


Figure A.2: Resulting Recipe Instruction Tree: Yaki Udon

squash and garlic on a baking tray and drizzle with the olive oil. Roast for 35-40 mins until soft. Season. Cook the pasta according to pack instructions. Drain, reserving the water. Use a stick blender to whizz the squash with 400ml cooking water. Heat some oil in a frying pan, fry the sage until crisp, then drain on kitchen paper. Tip the pasta and sauce into the pan and warm through. Scatter with sage."[25] The resulting recipe instruction tree is presented in figure A.3.

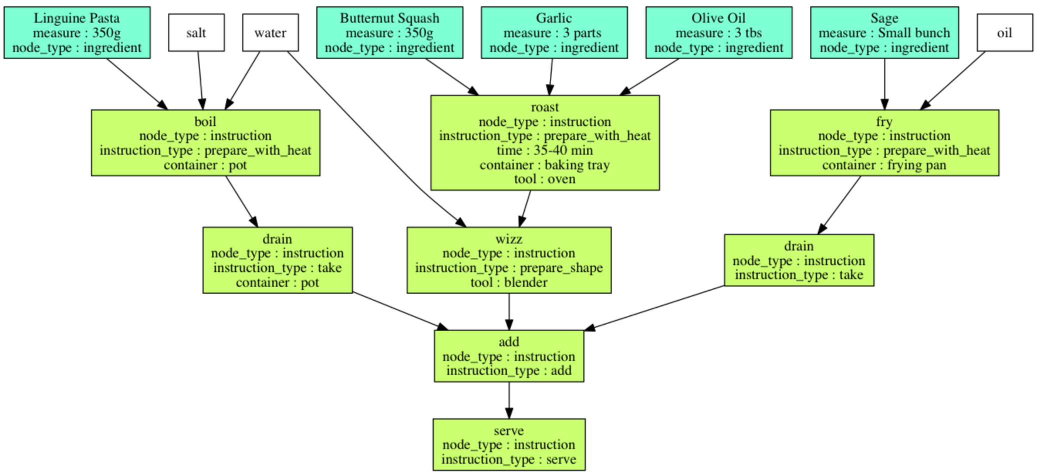


Figure A.3: Resulting Recipe Instruction Tree: Squash Linguine

Laksa King Prawn Noodles The ingredient list is: *1 tbsp Olive Oil, 1 finely sliced red chili, 2 1/2 tbsp Thai red curry paste, 1 vegetable stock cube, 400ml can coconut milk, 2 tsp fish sauce, 100g rice noodles, 2 juice of 1, the other halved lime, 150g king prawns, 1/2 small pack coriander* The instructions of Laksa King Prawn Noodles are: "*Heat the oil in a medium saucepan and add the chilli. Cook for 1 min, then add the curry paste, stir and cook for 1 min more. Dissolve the stock cube in a large jug in 700ml boiling water, then pour into the pan and stir to combine. Tip in the coconut milk and bring to the boil. Add the fish sauce and a little seasoning. Toss in the noodles and cook for a further 3-4 mins until softening. Squeeze in the lime juice, add the prawns and cook through until warm, about 2-3 mins. Scatter over some of the coriander. Serve in bowls with the remaining coriander and lime wedges on top for squeezing over.*"[24] The resulting recipe instruction tree is presented in figure A.4.

Pad See Ew The ingredient list is: *6oz/180g rice stick noodles, 2 tbsp dark soy sauce, 2 tbsp oyster sauce, 2 tsp soy sauce, 2 tsp white vinegar, 2 tsp sugar, 2 tbsp water, 2 tbsp peanut oil, 2 cloves garlic, 1 cup Chicken, 1 Egg, 4 cups Chinese broccoli* The instructions of Pad See Ew are: "*Mix Sauce in small bowl. Mince garlic into wok with oil. Place over high heat, when hot, add chicken and Chinese broccoli stems, cook until chicken is light golden. Push to the side of the wok, crack egg in and scramble. Don't worry if it sticks to the bottom of the wok - it will char and which adds authentic flavour. Add noodles, Chinese broccoli leaves and sauce. Gently mix together until the noodles are stained dark and leaves are wilted. Serve immediately!*"[23] The resulting recipe instruction tree is presented in figure A.5.

A.2 Evolutionary Algorithm Cycle - Intermediate Results

The Algorithm constructs trees which are sometimes to large to printed in the main chapters. Here are some real world example directly depicted from the algorithms output.

Recombination The Recombination merges parts of two instruction trees. Further explanation can be found in chapter 4.8. In the figure A.8, the first recombination is shown. The baseline recipes are *Spicy Arrabiata Penne* (see figure A.1) and *Pad See Ew*(see figure A.5). From the main recipe *Spicy Arrabiata Penne* a subtree is cut of and the remaining tree (see figure

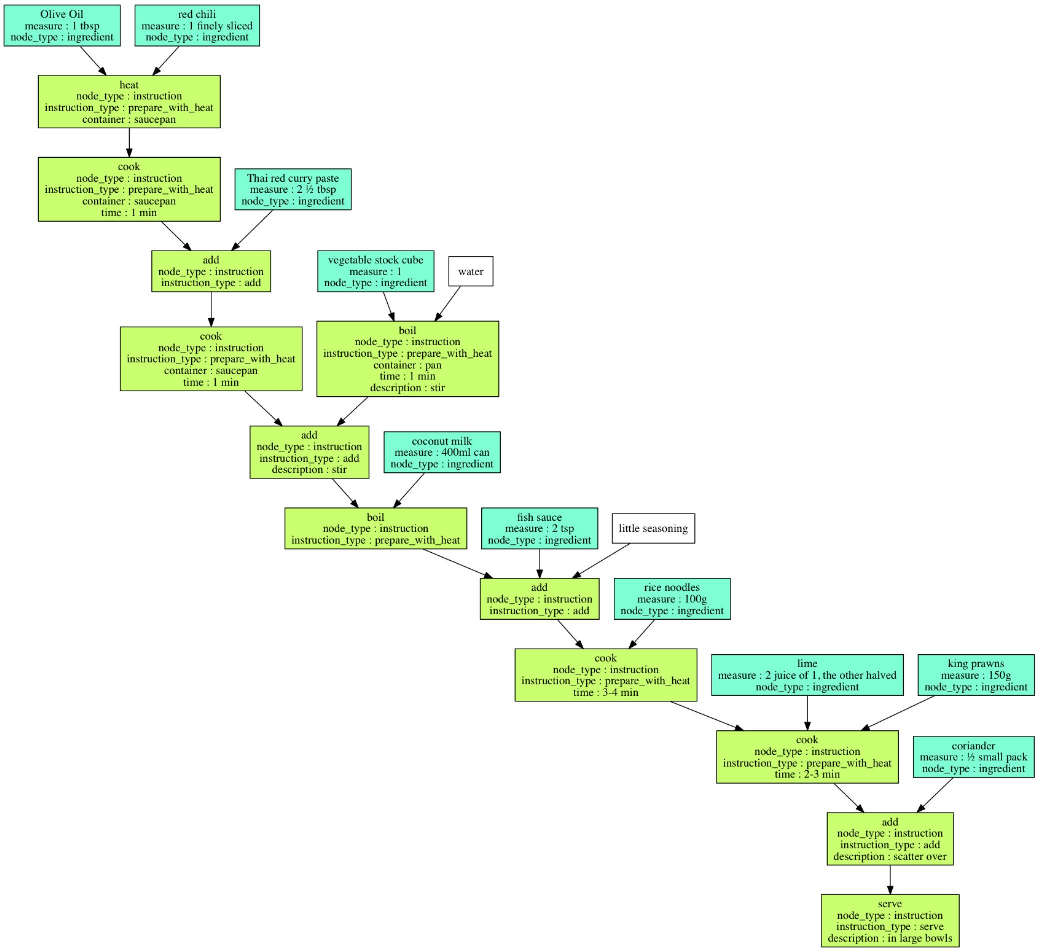


Figure A.4: Resulting Recipe Instruction Tree: Laksa King Prawn Noodles

A.6) build the main recipe tree. From the second parent recipe *Pad See Ew* a similar sized subtree selected from a set of possible sub trees and is cut of(see figure A.7). this subtree replaces the cut of subtree from parent recipe one (Spicy Arrabiata Penne).

Mutation In the Mutation one Random ingredient from the novel created recipes is substituted. In the the example recipe shown in figure A.8, the ingredient *peanut oil* was replace by *sweet almond oil.*

Postprocessing The postprocessing constructs Human Readable Recipes. The Resulting Recipe is Shown in figure A.10. These Recipes are fully automatic created and stored as html files.

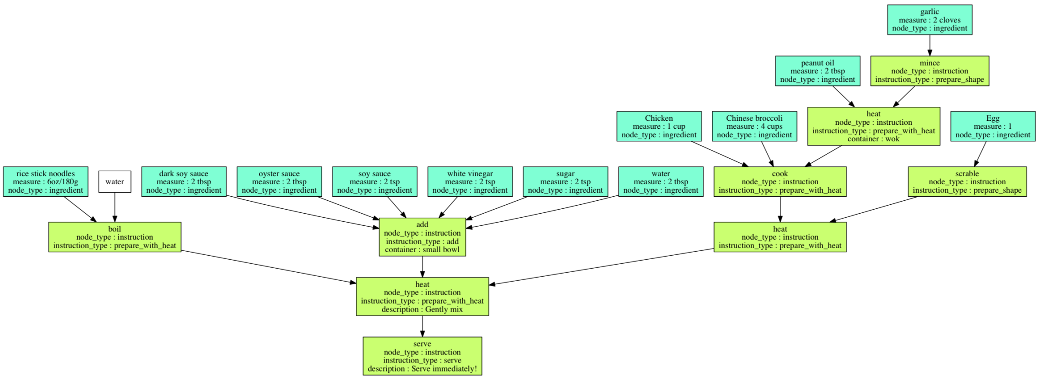


Figure A.5: Resulting Recipe Instruction Tree: Pad See Ew

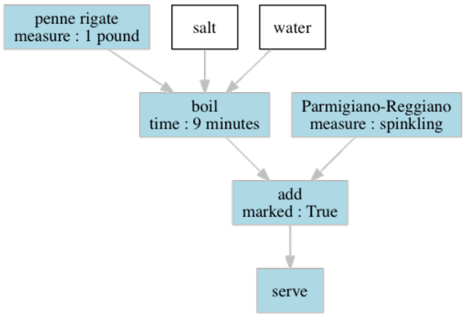


Figure A.6: Main Recipe Instruction Tree for Recombination

A.3 Published Poster

During the work at this Master Thesis we published the main approach at the Conference EvoStar 2019 in Leipzig. Figure A.11 shows the presented poster.

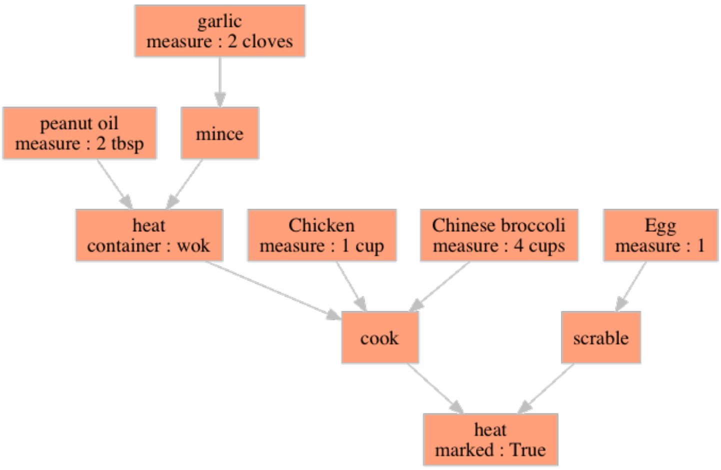


Figure A.7: Cut of Recipe Instruction Sub Tree for Recombination

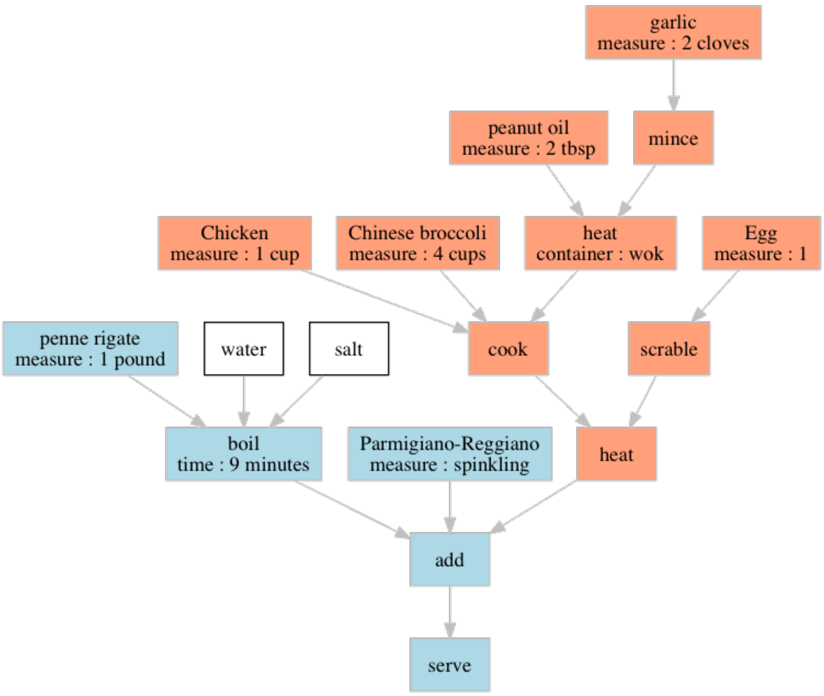


Figure A.8: Resulting Recipe Instruction Tree after Recombination

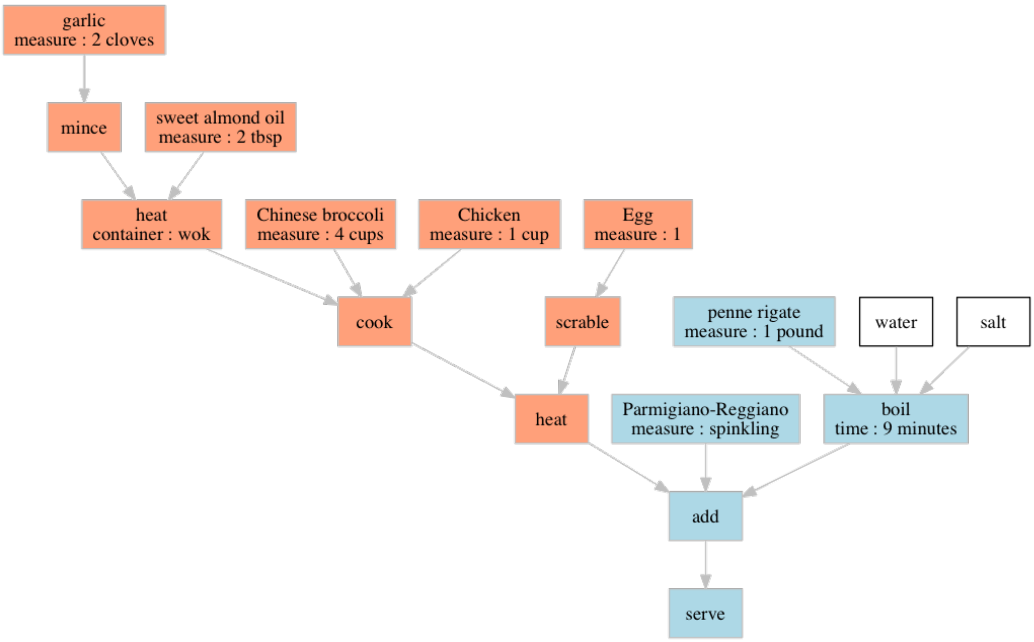


Figure A.9: Recipe with mutated ingredient based on recipe from figure A.8

Noodles with Chinese broccoli

Ingredient	Measure
garlic	2 cloves
sweet almond oil	2 tbsp
Chinese broccoli	4 cups
Chicken	1 cup
Egg	1
penne rigate	1 pound
Parmigiano-Reggiano	spinkling

Hint: The recipe is grouped in 5 major groups of tasks. The included subtasks should always be directly combined or mixed together.

1) To prepare another partial recipe component:
1.1: Boil 1 pound penne rigate, water and salt for 9 minutes.

2) For the dish you also need:
2.1: Mince 2 cloves garlic, next you...
2.2: Heat 2 tbsp sweet almond oil in a wok, then...
2.3: Cook 4 cups Chinese broccoli and 1 cup Chicken.

3) Next Component is:
3.1: Scrable 1 Egg.

4) Combine all previous results:
4.1: Heat.

5) Combine all previous results:
5.1: Add spinkling Parmigiano-Reggiano, followed by...
5.2: Serve

Figure A.10: Resulting Human Readable Recipe based on Recipe from figure A.9

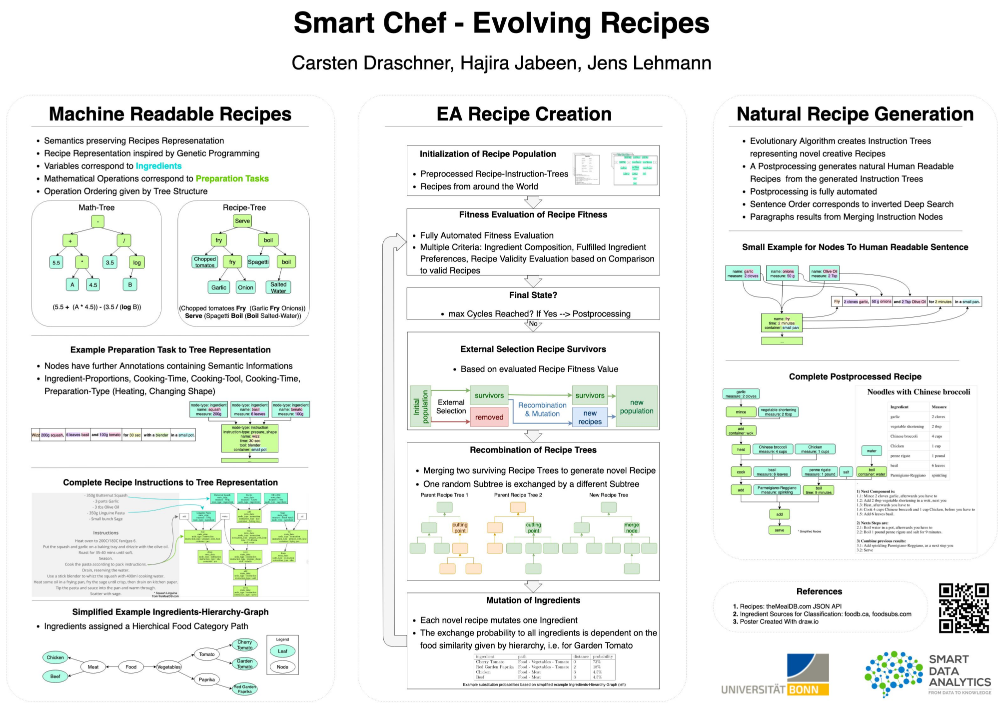


Figure A.11: Poster of Master Thesis Approach which was published at EvoStar 2019 Conference

Bibliography

[1] "Pandas." Accessed: 2019-03-11.

[2] "Is artificial intelligence set to become art's next medium?." `https://www.christies.com/features/A-collaboration-between-two-artists-one-human-one-a-machine-9332-1.aspx`. Accessed: 2018-01-21.

[3] C.-Y. Teng, Y.-R. Lin, and L. A. Adamic, "Recipe recommendation using ingredient networks," in *Proceedings of the 4th Annual ACM Web Science Conference*, pp. 298–307, ACM, 2012.

[4] "The science behind foodpairing." `https://www.foodpairing.com/en/science-behind`. Accessed: 2018-01-21.

[5] Y.-Y. Ahn, S. E. Ahnert, J. P. Bagrow, and A.-L. Barabási, "Flavor network and the principles of food pairing," *Scientific reports*, vol. 1, p. 196, 2011.

[6] "How ibm's chef watson actually works - bon appetit." `https://www.bonappetit.com/entertaining-style/trends-news/article/how-ibm-chef-watson-works`. Accessed: 2018-01-21.

[7] A. Elgammal, B. Liu, M. Elhoseiny, and M. Mazzone, "Can: Creative adversarial networks, generating" art" by learning about styles and deviating from style norms," *arXiv preprint arXiv:1706.07068*, 2017.

[8] "Chef watson has arrived and is ready to help you cook." `https://www.ibm.com/blogs/watson/2016/01/chef-watson-has-arrived-and-is-ready-to-help-you-cook/`. Accessed: 2018-01-21.

[9] "Foodb.ca." `foodb.ca`. Accessed: 2018-01-21.

[10] "Tree-based genetic programming." `https://developer.yummly.com/policies`. Accessed: 2019-03-10.

[11] "Bigoven." http://api2.bigoven.com/web/documentation/terms-of-use. Accessed: 2019-03-10.

[12] "themealdb," 2016-2019. Accessed: 2019-03-10.

[13] D. Cope, *Computer models of musical creativity*. MIT Press Cambridge, 2005.

[14] E. Cromwell, J. Galeota-Sprung, and R. Ramanujan, "Computational creativity in the culinary arts," in *FLAIRS Conference*, pp. 38–42, 2015.

[15] "We put a computer in charge of our test kitchen for a day, and here's what happened." https://www.bonappetit.com/test-kitchen/inside-our-kitchen/article/chef-watson-in-the-ba-test-kitchen. Accessed: 2018-01-21.

[16] "Cover:cheese." https://covercheese.appspot.com/. Accessed: 2018-01-21.

[17] "schema.org - recipe schema." Accessed: 2018-01-21.

[18] "recipesource.com." https://www.recipesource.com/holiday/00/rec0002.html. Accessed: 2019-03-10.

[19] "Ffts." http://www.ffts.com/recipes.htm. Accessed: 2018-01-21.

[20] "Edamam." https://developer.edamam.com/edamam-recipe-api. Accessed: 2019-03-10.

[21] M. Sam, A. Krisnadhi, C. Wang, J. Gallagher, and P. Hitzler, "An ontology design pattern for cooking recipes - classroom created," vol. 1302, 10 2014.

[22] "Foodsubs - the cook's thesaurus." http://www.foodsubs.com/. Accessed: 2018-01-21.

[23] "themealdb recipe padseeew." Accessed: 2019-04-01.

[24] "themealdb recipe laksakingprawnnoodles." Accessed: 2019-04-01.

[25] "themealdb recipe squashlinguine." Accessed: 2019-04-01.

[26] "themealdb recipe yakiudon." Accessed: 2019-04-01.

[27] "themealdb recipe spicyarrabiatapenne." Accessed: 2019-04-01.

[28] K.-J. Kim and C.-H. Chung, "Tell me what you eat, and i will tell you where you come from: A data science approach for global recipe data on the web," *IEEE Access*, vol. 4, pp. 1–1, 01 2016.

[29] G. Finnveden, M. Z. Hauschild, T. Ekvall, J. GuinÃ©e, R. Heijungs, S. Hellweg, A. Koehler, D. Pennington, and S. Suh, "Recent developments in life cycle assessment," *Journal of Environmental Management*, vol. 91, no. 1, pp. 1 – 21, 2009.