Subs and Sandwiches: Replacing One Ingredient By Another

Henry Larkin and Derek Bridge*

Insight Centre for Data Analytics,
School of Computer Science and Information Technology,
University College Cork, Ireland
h.larkin@umail.ucc.ie|derek.bridge@insight-centre.org

Abstract. The Earl system works in the domain of sandwich recipes. We show how Earl uses two sources of knowledge —ingredients knowledge that is extracted from a gastronomy web site, and a case base of sandwich recipes— to propose substitutes for ingredients in sandwich recipes. In particular, we describe four substitution methods: human-authored substitutions, taxonomically-driven substitutions, lexically-driven substitutions, and frequent item set substitutions. A preliminary experiment with human participants shows that a system that uses the last of these methods is competitive with one that uses the other three methods in combination.

1 Introduction

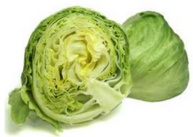
Over the years, the Computer Cooking Contest has hosted many systems that substitute one ingredient for another in a recipe. In the JaDaCook case-based system [6], for example, the user submits her desired ingredients; JaDaCook then retrieves from its case base a set of recipes that contain as many of these ingredients as possible; but, where one of the user's desired ingredients is not present in a recipe that it has retrieved, it adapts the recipe by substituting the desired ingredient for the most similar ingredient in the recipe.

JaDaCook makes substitutions in order to insert ingredients that the user desires but which are not in the recipe. But substitutions can be made for other reasons. For example, a system might allow the user to replace undesirable ingredients, e.g. foodstuffs that she does not like, ones that she cannot eat for medical or cultural reasons, or ones that are not available. The third version of the CookIIS system handles forbidden ingredients in this way [7].

In this paper, we look at two sources of knowledge that a system can use to propose substitutes for an ingredient. The first is a remarkable web resource: the Cook's Thesaurus.[1] It provides us with three ways to make substitutions in recipes: human-authored, taxonomically-driven and lexically-driven. The second

* This publication has emanated from research supported in part by a research grant from Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289.

[1] www.foodsubs.com



iceberg lettuce = head lettuce = cabbage lettuce = crisphead lettuce ***Notes:*** This is prized for its crispness and longevity in the refrigerator, but it's a bit short on flavor and nutrients. ***Substitutes:*** romaine lettuce (also crunchy, and more flavorful) OR leaf lettuce

Fig. 1: Example extract from the Cook's Thesaurus

is a large recipe case base that we have scraped from the web. From it, we define a fourth way to make substitutions: frequent item set substitutions.

All the recipes in our case base are for sandwiches. But our substitution methods are quite general. We have implemented all four methods in a system called Earl, named after John Montagu the 4th Earl of Sandwich.

Section 2 describes the Cook's Thesaurus; Section 3 describes the sandwich recipe case base; Section 4 describes the four substitution methods; Section 5 reports a simple experiment; Section 6 considers avenues for future inquiry.

2 The Cook's Thesaurus

The Cook's Thesaurus is a gastronomy web site. Its home page links to pages that describe categories, e.g. Vegetables, Fruits and Dairy. A category page has links to pages that describe sub-categories, e.g. Citrus Fruit, Berries and Stone Fruit. A sub-category page contains entries that describe ingredients, e.g. lemon, lime and limequat. In some cases, there is an extra level of hierarchy, e.g. the Dairy category links to the Cheese sub-category, which links to sub-sub-categories that include Soft Cheese and Firm Cheese, before one reaches pages of actual cheeses, such as Boursault and Brie.

The description of a specific ingredient typically contains some or all of the following: its name; a guide to pronouncing its name; alternative names; notes (a descriptive text); and suggested substitutes that can be used in cooking. Figure 1 shows the description for iceberg lettuce.

We scraped this web site and stored the extracted data in a database. We obtained over 1500 items (categories, sub-categories, and ingredients). In Earl, we make use of the ingredient names, the descriptive text, the substitutes, and the hierarchy of categories, sub-categories and ingredients. Although Earl uses only sandwich ingredients, the Cook's Thesaurus covers ingredients used in all aspects of cooking, not just those relevant to sandwiches, and so it could be a useful resource to other entrants to the Computer Cooking Contest.

3 The Sandwich Case Base

Earl uses case-based reasoning to create sandwiches. To build Earl, we needed a case base of sandwich recipes. We scraped approximately 10,000 sandwich recipes from five web recipe web sites. We stored each recipe's name and its list

Ingredient	**Human-authored substitutes**	**Taxonomically-driven substitutes**	**Lexically-driven substitutes**
Bell pepper	Italian frying pepper	Pimento	Romaine lettuce
Swiss cheese	Emmentaler	Caciotta cheese	Teleme cheese
Jalapeño pepper	Fresno chili	Serrano pepper	Pear
Apple	Pear	Pear	Lard
Onion	White onion	Shallot	–

Table 1: Substitutions from the Cook's Thesaurus

of ingredients. There is noise in the naming of ingredients on recipe web sites. For example, sandwich recipe authors may misspell "mayonnaise" or even use a brand name, such as "Hellmann's", instead. To give a canonical representation of each ingredient, we linked each sandwich ingredient to one ingredient in the database that we had scraped from the Cook's Thesaurus. We provide more details of how we built Earl's case base in [4].

4 Making Substitutions

In this section, we present Earl's four methods for suggesting substitutes.

4.1 Human-authored substitutions

As we have explained, for some ingredients, the Cook's Thesaurus explicitly lists substitutes and we have extracted these for use in our system. A domain expert has written these, and so we consider them to be very reliable. Figure 1, shows the substitutes for iceberg lettuce to be romaine lettuce and leaf lettuce. The second column of Table 1 gives more examples.

The obvious limitation is that not every ingredient has human-authored substitutes: only 81% do. A more subtle problem is that many of these substitutes are just variants of the ingredient. This is good if there is a particular reason a user does not want to have a specific ingredient in the sandwich, such as not having that ingredient on hand, but it is less helpful if the user has an aversion to that ingredient and ones like it. For example, the substitutes for 'onion' are 'yellow onion', 'white onion' and 'shallot'. If the user outright dislikes the taste of onions, none of these suggestions is likely to be any more palatable.

4.2 Taxonomically-driven substitutions

The Cook's Thesaurus is organised hierarchically. Ingredients that are similar gastronomically are close together in the tree structure. Distance through the hierarchy denotes a form of gastronomic similarity. Similar ingredients (ones that are close in the hierarchy) can be substituted for one another. Numerous entrants to the Computer Cooking Contest use variants of this idea, in some cases using taxonomies (e.g. [6, 8]), in other cases using formal ontologies (e.g.

the one used in the Taaable system [3]). For a taxonomy or ontology to be useful in the Computer Cooking Contest, it is important that it organizes ingredients gastronomically and not biologically. For example, lettuce and arugula (rocket) should be close in the hierarchy: when put in a sandwich, both fulfil the role of tasty green somewhat crunchy thing. But they are less similar genetically: lettuce is of the family asteraceae while arugula is of the family brassicaceae. The Cook's Thesaurus is ideal in this regard, as evidenced by the fact that Taaable's food ontology is also inspired by it [3].

When using this method to propose substitutions, Earl takes the very simplest approach: it suggests replacing an ingredient only by its siblings in the tree. The third column of Table 1 gives examples.

This way of proposing substitutions suffers from some of the same problems as the human-authored ones. Some ingredients, those that do not have siblings, will not have any substitutions. Of course, we could look further across the taxonomy, e.g. at cousins, but we chose not to do this. We felt that cousins did not preserve sufficient gastronomic similarity; in JaDaCook, Herrera and Iglesias took the same decision for the same reason [6].[2] But this does mean that, as with the human-authored substitutions, taxonomically-driven substitutions are often just variants of the original ingredient, which may not always satisfy the user.

4.3 Lexically-driven substitutions

As we have seen, human-authored substitutions and taxonomically-driven substitutions (at least when confined to siblings) have the problem that substitutions may be too similar to the original ingredient. Lexically-driven substitutions have the potential to overcome this problem.

For many ingredients, the Cook's Thesaurus gives a short description, henceforth referred to as a *titbit*. Titbits often describe taste and texture. Earl can propose to replace an ingredient by another ingredient having a lexically similar titbit. This results in substitutions of ingredients that have lexically similar titbits but which might not be close in the taxonomy.

For each titbit, we used a part-of-speech tagger to pick out the adjectives. We constructed a vector space representation taking these adjectives as terms and using tf-idf weighting. To propose lexically-driven substitutions, Earl uses cosine similarity on the vector representations of the titbits. The fourth column of Table 1 gives examples.

Lexical similarity and term vector representations are not new to the Computer Cooking Contest. For example, the *What's in the Fridge?* system uses cosine similarity between the query and the text of a recipe for case retrieval [8]; and RaGoÛt represents recipes by term vectors [1]. But, to the best of our

[2] We are aware that this decision ignores the possibility that, in some parts of the taxonomy, cousins may be sufficiently similar to be substitutes for each other. For example, if there is more detail (e.g. greater depth) to, say, the subtree of fruits than there is in the subtree of meats, then substitution of a fruit by its cousin may be acceptable in a way that substitution of a meat by its cousin is not.

knowledge, previous entrants have not made use of ingredient descriptions, and so this is a new substitution method.

To the extent that titbits describe the gastronomic properties of an ingredient (e.g. “crunchy”, “chewy”, “savoury”), we hoped this method would find substitutions that are not mere variants but are gastronomically appropriate. In practice, at times the suggestions are good (e.g. romaine lettuce for bell pepper); sometimes they seem a little odd but may be worth trying (e.g. pear for jalapeño pepper); and other times they are very poor (e.g. lard for apple). Titbits often contain historical information, references to other ingredients, or digressions. In these cases, the method matches on irrelevant adjectives, giving poor results.

We could avoid some of the most inappropriate substitutions by removing from the taxonomy any ingredient from the Cook’s Thesaurus that is rarely used in sandwiches (e.g. lard). We chose not to do this both because of the manual effort involved and because it felt like cheating.

4.4 Frequent item set substitutions

Our final method is different from those above. They focus on replacing an ingredient with one that has some similarities. Here, the driving idea is to find an ingredient that goes well with everything else in the sandwich. We do this by analysing frequent item sets generated from the recipe case base.

In Earl, a frequent item set is a set of ingredients that occurs often across the recipe case base. We find frequent item sets using the Apriori algorithm [2], and we define frequent as appearing in at least 5% of the recipes.

To propose a substitution of an ingredient in a sandwich, we obtain the residual ingredients, i.e. the ingredients of the sandwich other than the ingredient that is to be replaced. We then find the frequent item set that has two properties: (a) it is the one that overlaps most with the residual ingredients[3], and (b) it has at least one additional ingredient. Property (a) means that this frequent item set is similar to the sandwich; property (b) gives us a target ingredient (or ingredients) to substitute for the ingredient that is to be replaced. We know that the new ingredient ‘goes well’ with at least those ingredients that the sandwich shares with the frequent item set.

The second column of Table 2 gives examples when running this method on a sandwich that contains: bread, butter, bell pepper, Swiss cheese, apple, onion, jalapeño pepper and ham. It is important to note that, for this method (but not for any of the others), changing the sandwich will produce different substitutions. The third column of Table 2 gives example substitutions when the original sandwich contains beef instead of ham.

This method does not suffer from the weakness of the other methods, namely that they propose substitutions that are too similar to the original. On the other hand, its weakness is that, since it can suggest any other ingredient in the case base, sometimes it suggests something that is too different from the original.

[3] Ties, which are rare, are broken randomly.

Ingredient	Suggested substitutes in a ham sandwich	Suggested substitutes in a beef sandwich
Bell pepper	Pickle	Olives
Swiss cheese	Pickle	Olives
Jalapeño pepper	Lettuce	Tomato
Apple	Garlic	Mayonnaise
Onion	Mayonnaise	Olives

Table 2: Frequent item set substitutions from our recipe case base. The sandwich originally contains bread, butter, bell pepper, Swiss cheese, apple, onion, jalapeño pepper and either ham or beef.

Earl is not the first system to try to extract knowledge from its recipes. For example, in adding or removing an ingredient from a recipe, Taaable 4 seeks variation regularities across similar recipes to the one that is being adapted, in order to propose additional insertions to, or deletions from, the recipe [5]. To the best of our knowledge, our use of frequent item sets is new.

5 A Preliminary Experiment

We have run a simple online experiment to obtain a preliminary evaluation of these substitution methods. It was not practical to recruit enough participants with enough patience to evaluate each method against each other. Instead, in this preliminary experiment, we asked participants to compare frequent item set substitutions with a combination of the other methods.

In advance, we selected nine sandwiches from the recipe case base. We randomly selected an ingredient in each sandwich. We computed a substitute using a 'cascade' of the methods that use the Cook's Thesaurus. The cascade places what we expect to be reliable methods ahead of less reliable ones: human-authored, then taxonomically-driven, and lastly lexically-driven. In other words, if we could not find a human-authored substitute, we ran the taxonomically-driven method; and if this failed to find a substitute, we resorted to the lexically-driven method. Henceforth, we refer to this substitute as the Cook's Thesaurus substitute (CT for short). We also computed a frequent item set substitute (FIS for short).

The web page displayed a sandwich, the ingredient to be replaced, and the proposed substitutes (CT and FIS). We asked participants to rate both substitutes on a scale of 1 to 10, with 1 being low and 10 being high. What we wanted the participants to tell us was whether the substitution was appropriate or not, irrespective of their personal tastes. Pilot testing revealed that many people had difficulty being objective in this way. If someone did not like ham and the substitute was ham, participants awarded the substitute a low rating, regardless of whether the substitution was appropriate or not. Therefore, in the experiment itself, we tried to encourage participants to be objective by asking for two ratings for each substitute: a personal rating (where they could tell us about their preferences) and an impersonal rating (where we hoped they would

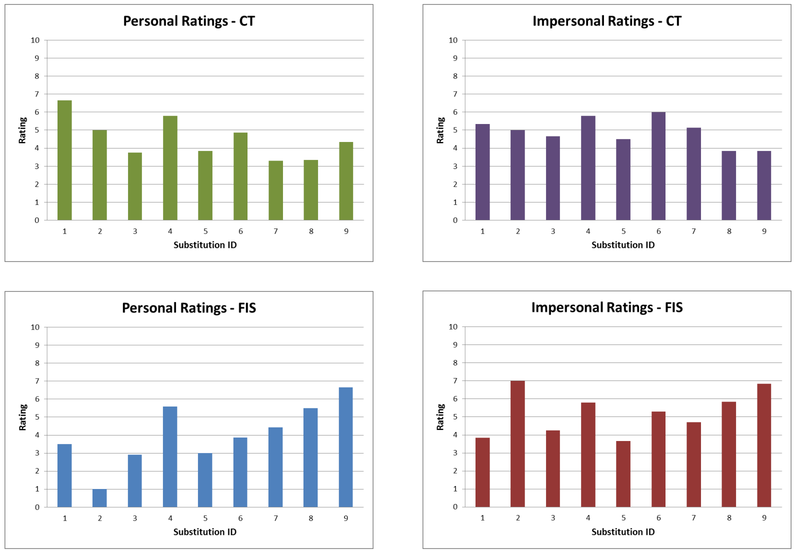


Fig. 2: Average ratings by substitution

be more objective). We hoped that, by asking both questions, we would make participants more aware of the difference between them. The results we are most interested in are the impersonal ones since Earl does not have user profiles and therefore cannot take the user's personal preferences into account.

A total of 56 people completed the experiment. Hence, each of the 9 pairs of substitutes was shown to at least 6 participants. Figure 2 shows the average rating each substitution received by method (CT or FIS) and rating type (Personal or Impersonal). There is a lot of variance, especially in the personal ratings. The CT substitutions have less variance than the FIS ones. If we focus on the impersonal ratings, we see that none of the CT substitutions is rated quite as poorly as the worst FIS substitutions, but likewise they do not rate as highly as the best FIS substitutions.

There are some oddities. For example, most participants seem to dislike FIS substitution number 2 (low personal rating) but they reason that it is actually a good substitution (highest impersonal rating). This substitution proposes olives. It seems that many of our participants simply do not like olives.

CT substitutes are better on average in the personal ratings (4.5 for CT and 4.1 for FIS), but the reverse is true in the impersonal ratings (4.9 for CT but 5.2 for FIS). But the differences are not statistically significant at even the 10% level (using a paired, two-tailed t-test).

6 Conclusions

Our evaluation surprised us! Frequent item set substitutions are competitive with ones that use knowledge from the Cook's Thesaurus. This conclusion comes with the caveat that the experiment is of very small scale. Furthermore, the majority of participants are 18–23 year old students. Participants with different demographics might draw on different experiences or have different tastes.

There are many avenues for future work. We could begin by improving the lexically-driven substitutions. In this regard, we could try to confine the term vector representation to adjectives that describe the taste and texture of ingredients, avoiding less relevant adjectives.

We could devise further methods that make use of the nutritional information, or places of origin or dietary restrictions in Taaable's semantic wiki.[4]

In our simple experiment, CT substitutions result from a cascade of three methods. Outside of this experiment, Earl uses a cascade of all four methods. It would be interesting to consider ways of combining methods other than using a cascade, e.g. some form of weighted vote.

All four substitution methods that we have described apply to recipes in general; they are not confined to sandwich recipes. However, we recognize that additional factors come into play when making substitutions in cooking recipes; for example, the chemistry of the substitute must resemble the original ingredient if the cooking is to succeed.

References

1. Ibrahim Adeyanju, Susan Craw, Abhishek Ghose, Allyson Gray, and Nirmalie Wiratunga. RaGoÛt: An arpeggio of tastes. In *Workshop Programme of the 9th European Conference on Case-Based Reasoning*, pages 229–238, 2008.
2. Rakesh Agrawal and Ramakrishman Srikant. Fast algorithms for mining association rules. In *Procs. of the 20th International Conference on Very Large Data Bases*, pages 487–499, 1994.
3. Fadi Badra et al. TAABLE: Text mining, ontology engineering, and hierarchical classification for textual case-based cooking. In *Workshop Programme of the 9th European Conference on Case-Based Reasoning*, pages 219–228, 2008.
4. Derek Bridge and Henry Larkin. Creating new sandwiches from old. In *Workshop Programme of the 22nd International Conference on Case-Based Reasoning*, (this volume) 2014.
5. Julien Cojan, Valmi Dufour-Lussier, Emmanuelle Gaillard, Jean Lieber, Emmanuel Nauer, and Yannick Toussaint. Taaable 4: Knowledge extraction for improving case retrieval and recipe adaptation. In *Workshop Programme of the 19th International Conference on Case-Based Reasoning*, pages 197–206, 2011.
6. P. Javier Herrera and Pablo Iglesias. JaDaCook: Java application developed and cooked over ontological knowledge. In *Workshop Programme of the 9th European Conference on Case-Based Reasoning*, pages 209–218, 2008.
7. Régis Newo, Kerstin Bach, Alexandre Hanft, and Klaus-Dieter Althoff. On-demand recipe processing based on CBR. In *Workshop Programme of the 18th International Conference on Case-Based Reasoning*, pages 209–218, 2010.
8. Qian Zhang, Rong Hu, Brian Mac Namee, and Sarah Jane Delany. Back to the future: Knowledge light case base cookery. In *Workshop Programme of the 9th European Conference on Case-Based Reasoning*, pages 239–248, 2008.

[4] `wikitaaable.loria.fr`