RaGoÚt: An Arpeggio of Tastes*

Ibrahim Adeyanju¹, Susan Craw¹, Abhishek Ghose², Allyson Gray¹, and Nirmalie Wiratunga¹

¹ School of Computing, The Robert Gordon University, Aberdeen, Scotland
² Department of Computer Science and Engineering, IIT Madras, India

Abstract. This paper presents the RaGoÚt system developed in response to the ECCBR’08 Computer Cooking Contest call. It proposes a case authoring approach that combines domain-specific and independent knowledge sources to create a feature vector representation of recipes. Case retrieval uses semantic similarity knowledge acquired from WordNet. This is combined with exact matching to enforce adaptation-aware case ranking, followed by substitutional adaptation of mismatched query ingredients. RaGoÚt addresses all outlined CCC tasks: compulsory, negation and menu challenges. Preliminary results from the compulsory and negation tasks are very favourable for both retrieval and adaptation. Although RaGoÚt can also retrieve three-course menus in response to the third menu task, its adaptation strategy for menus is being improved.

1 Background

The RaGoÚt system builds on RGU’s Case-Based Reasoning (CBR) research on feature extraction from text [1], case retrieval nets for text [2] and adaptation [3], and benefits from IIT Madras’ research on the extraction of word cohesion relations [4]. RaGoÚt was designed and implemented initially as an honours project and completed as a research project. This effort continues our participation in the Textual CBR workshop’s air traffic incident report challenge [5].

Extraction of ingredients and relevant indexing terms from text content is discussed next in Section 2. Case representation and its generalisation using similarity knowledge appears in Section 3. This is followed by our retrieval and reuse strategy in Section 4. A functional overview of the RaGoÚt system appears in Section 5 with initial retrieval results and conclusions in Section 6.

2 Case Representation

The provided recipe file is a semi-structured textual document where each recipe has a recipe title labeled TI, a list of ingredients each labeled IN, and a single preparation part labeled PR. Our case representation extracts terms from the text for a recipe so that these terms can be used as an index to the recipes.

* This joint research is supported by a UKIERI standard award to the Robert Gordon University and IIT Madras.
The main content of a query is often specific ingredients. The challenge here is that words in the recipes may be redundant for recipe matching, such as teaspoon, but they may also be too specialised, such as halibut rather than fish, and adaptation by ingredient substitution must apply relationships between ingredients. The non-ingredient query terms are cuisine types, dietary practices and meal types. These may be explicitly found in any part of the recipe: TI(tle), IN(gredients) and PR(eparation). When these non-ingredient terms fail to be explicitly mentioned in the recipes, associations between recipes and terms are inferred by using the web as a source of background knowledge.

2.1 Pre-Processing of Text

The first step is to remove the stopwords that provide little meaning for the recipe retrieval task. In addition to the normal stopwords of English text such as punctuation, “a”, “the”, “and”, . . . , there are recipe-specific stopwords that are not useful for selecting a similar recipe. Common cooking units of measure such as gram, pint and cup are also removed as stopwords, as are numbers specifying any quantities. Similarly the state of an ingredient, such as chopped, grated or sliced, is regarded as not meaningful for the retrieval task. As an example, the IN(gredient) 1 cup packed brown sugar would be reduced to brown sugar during pre-processing. As a final pre-processing step a very simple stemming is applied that turns plurals into their singular form.

2.2 Ingredient Extraction from Recipe Documents

The TI(tle), IN(gredient) and PR(eparation) parts of a recipe are scanned to identify unigrams and bigrams that may be ingredient terms. The example above would generate brown, brown sugar, and sugar as potential ingredient terms.

We use WordNet [6] as the first knowledge source to identify a term as an ingredient. The hypernym trees for the term are retrieved from WordNet and checked to see if the keyword food is a hypernym. Figure 1 shows the approach for chicken.

However, not all valid ingredients have food as a hypernym; e.g. Figure 2 shows the relevant hypernym tree for nut. To accommodate this, three other food-related keywords were identified: fruit, leaven (e.g. baking powder, yeast), and substance (e.g. baking soda). These keywords are used in the same way as food when checking the hypernym trees.

2.3 Extraction of Other Features

The cuisine types are identified in the recipe bigrams and unigrams by scanning their WordNet hypernym trees for key words related to cuisine. Using Chinese as a model we discovered some useful keywords: person, nation, land, country and natural language. Thus we identified 47 cuisine type and 19 meal type terms in the recipes in the same way as we identified ingredient terms. These extracted ingredient types and meal types are explicitly listed as domain knowledge together.
chicken, poulet, volaille -- (the flesh of a chicken used for food)
  => poultry -- (flesh of chickens or turkeys or ducks or geese raised for food)
  => bird, fowl -- (the flesh of a bird or fowl (wild or domestic) used as food)
  => meat -- (the flesh of animals (including fishes and birds and snails) used as food)
  => food, solid food -- (any solid substance (as opposed to liquid) that is used as a source of nourishment; “food and drink”)
  ...
  => entity -- (that which is perceived or known or inferred to have its own distinct existence (living or nonliving))

Fig. 1. WordNet Hyponym Tree for Chicken

nut -- (usually large hard-shelled seed)
  => seed -- (a small hard fruit)
  => fruit -- (the ripened reproductive body of a seed plant)
  => reproductive structure -- (the parts of a plant involved in its reproduction)
  => plant organ -- (a functional and structural unit of a plant or fungus)
  => plant part, plant structure -- (any part of a plant or fungus)
  => natural object -- (an object occurring naturally; not made by man)
  ...
  => entity -- (that which is perceived or known or inferred to have its own distinct existence (living or nonliving))

Fig. 2. WordNet Hyponym Tree for Nut

with manually identified dietary practices and ingredient types The ingredient type terms include fruit, grain, meat, nut, oil, pasta, poultry, seafood, seed, vegetable. The meal type terms include appetizer, cake, dessert, soup, salad. The dietary practice terms include vegetarian but many are ingredients that must not be included such as nut free or non alcoholic. These negative requirements are handled by the query processing (see 4.3).

2.4 Web as Background Knowledge

We found that with a majority of recipes, both the cuisine and meal type terms were not explicitly included: 669 recipes have no explicit cuisine and 453 have no explicit meal type. In order to establish and label recipes with the most likely cuisine or meal type we used web search hits to determine the cohesion between a given recipe title and each of the candidate cuisines and meal types. For example, given a recipe cinnamon rolls, we first note the number of hits returned with a restricted Google search query. Here the restriction is enforced so that search is confined to documents related to cooking within Google’s cooking directory [7]. A
second query is constructed by combining both the recipe name and a candidate
cuisine (e.g. cinnamon rolls + Chinese). In order to establish a recipe’s cohesion
with either a cuisine or meal type term, we combine the number of hits returned
from both these queries as a ratio:

\[
\text{cohesion}(\text{recipe}, \text{term}) = \frac{\text{hits(\text{recipe} \& \text{term})}}{\text{hits(\text{recipe})}}
\]

We evaluated this web-based recipe-term association approach on a sample
of recipes containing explicit cuisine and meal type terms. The top 5 highest
cohesion terms associated the recipe with the correct term 80% of the time. As a
result of this initial experiment, any recipe that had no explicit cuisine (or meal
type) mentioned was associated with the top 6 cohesion terms. The degree of
association is a function of the cohesion measure and its rank.

3 From Recipe to Case

The recipe vocabulary consists of the ingredient and cuisine type terms extracted
from the recipes, together with the ingredient type, meal type and non-negative
dietary practice terms provided as domain knowledge.

3.1 Bag-of-Words Representation

As a first step, each recipe is transformed into a bag-of-words representation
of terms explicitly mentioned in the recipe. An additional generalisation step
extends the bag-of-words by activating the ingredient type terms that were used
to identify any ingredients explicitly mentioned in the recipe. The ingredient
type acts as a generalised term and multiple ingredient types may be identified
for an ingredient. For example Figure 1 enables a recipe that explicitly mentions
the ingredient chicken to also activate the terms poultry and meat.

Figure 3 illustrates the background knowledge used to transform the recipe
documents into a set of bags-of-words. The so far unused domain knowledge
containing Chinese, Indian, Thai and other cuisine type ingredients lists ingre-
dients distinctive of the particular cuisine. This knowledge further extends the
bag-of-words by activating the cuisine type term for any ingredient found in
its ingredient list that is explicitly listed in the recipe. For example an ingre-
dient macaroni would activate the cuisine type italian if macaroni were listed
as an Italian ingredient. These activations are further boosted with recipe-term
cohesion values obtained from Web querying.

The highlighted recipe in Figure 3 is the Macaroni and Chicken Casserole
recipe listed in Figure 4. Its bag-of-words is \{chicken, macaroni, milk, mushroom
soup, cream cheese, onion rings, poultry, meat, vegetable, italian\}. Finally, the
bag-of-words is represented as a feature vector of ingredient, ingredient type,
cuisine type, dietary practice and meal type terms in the vocabulary.

The feature vectors for each recipe are assembled into a recipe \times term matrix
\(R\) where each of the \(r\) rows corresponds to the feature vector for that recipe,
3.2 Semantic Relevance through WordNet

The matrix $R$ identifies whether terms are relevant or not for a recipe based on the terms explicitly mentioned in the recipe. We need to extend this representation to indicate the degree of relevance of a term for a recipe even in the absence of that term. For example a recipe that uses beef as an ingredient may be worth retrieving, and adapting, when a chicken dish is sought. Thus we wish to increase the zero entry for chicken in this recipe vector to capture the similarity of beef and chicken as ingredients.

It will be convenient to note that the vocabulary of $t$ terms comprises $i$ ingredient terms, $j$ ingredient type terms, $m$ meal type terms, $c$ cuisine terms and $d$ dietary terms.

A term x term similarity matrix $I$ is constructed for pairs of the $i$ ingredient terms. This matrix is symmetric and its leading diagonal contains ones. The off-diagonal entries contain the similarities between the pairs of terms.
<RECIPE>
<TI>Macaroni and Chicken Casserole</TI>
<IN>1 3 pound chicken, cooked and boned</IN>
<IN>2 c Uncooked macaroni</IN>
<IN>2 1/2 c Milk</IN>
<IN>2 cn Cream of mushroom soup</IN>
<IN>1 8 oz. Philadelphia cream cheese</IN>
<IN>1 cn French fried onion rings</IN>
<PR>Cook and bone chicken. Place chicken in bottom of 13 x 9 inch pan. Pour uncooked macaroni over chicken. Pour milk over macaroni. Spread soup over mixture. Cut cream cheese into small pieces and lay over soup. Cover and place in refrigerator overnight or at least 8 to 10 hours. One hour before baking, take out of refrigerator and let set. Heat oven to 350 degrees and bake for 50 minutes uncovered. Cover with onion rings and bake for 5 to 10 minutes more.</PR>
</RECIPE>

Fig. 4. Recipe for Macaroni and Chicken Casserole

similarity between a pair of ingredient terms $T_1$ and $T_2$ is calculated using Wu & Palmer’s similarity [8]:

$$I_{ij} = \frac{2 \cdot d(LCA)}{d(T_i)d(T_j)}$$

where $d(t)$ is the depth of term $t$ from the entity root in WordNet and LCA is the least common ancestor of the two ingredient terms. A similarity threshold of 0.7 is applied to limit the extent of the breadth first search of WordNet.

A term x term similarity matrix for cuisines $C$ is constructed in a similar fashion for the $c$ cuisines. A vocabulary-wide similarity matrix $S$ is constructed from these WordNet-populated similarity matrices $I$ and $C$, and the $1_j$, $1_m$, and $1_d$ identity matrices of size $j$, $m$ and $d$, as shown in Figure 5. The Relevance Recipe matrix $R'$ is generated from the binary Recipe matrix $R$ and the Similarity matrix $S$ as follows. We use a Max operator that replaces the sum in standard matrix multiplication. Thus

$$R'_{ij} = Max_{k=1}^t R_{ik} \cdot S_{kj}$$

Whereas $R$ is the incidence matrix of terms in the recipes, $R'$ captures the relative relevance of terms in recipes and is much less sparse than $R$. This generalisation forms the basis for fast case retrieval nets presented in [2].

4 Recipe Retrieval and Reuse

We shall focus first on queries without negation. These use the same vocabulary and binary feature vector representation as $R$. However the user interface offers only a subset of the vocabulary to allow a limited but useful choice of query terms. This is not a restriction of the approach but instead simplifies the choice of the user.
4.1 Retrieval

The matrices $R$ and $R'$ represent the case base of recipes without and with relative relevance knowledge. Two similarities $\text{sim}_R$ and $\text{sim}_R'$ are calculated between the query and the recipes represented in $R$ and $R'$ respectively. Each is computed using the dot product which emphasises the query terms that are matched, unlike the cosine similarity where length normalisation can adversely penalise exact matches when recipes are longer.

$\text{sim}_R$ retrieval over $R$ favours the explicit presence of query terms in the recipe. In addition, $\text{sim}_R'$ retrieval over $R'$ can favour the presence of recipe terms that are similar to the query terms and so ensures that recipes with ingredients that are similar to those requested are considered, even if the exact ingredients are absent. We arrive at a final recipe similarity by a weighted combination of $\text{sim}_R$ and $\text{sim}_R'$:

$$\text{sim}(\text{recipe}, \text{query}) = w \times \text{sim}_R(\text{recipe}, \text{query}) + (1 - w) \times \text{sim}_R'(\text{recipe}, \text{query})$$

The recipes are ranked according to this weighted similarity value. We have found that higher values for $w$ give best results for the CCC queries and so we use 0.8 as the weight.

4.2 Reuse

The 5 top ranked recipes are retrieved and each is considered for reuse. If the similarity indicates an exact match with the requirements in the query then the recipe is reused unchanged. Otherwise adaptation may be necessary.

Substitution adaptation is applied by finding where query mismatches occur and which recipe terms do not occur in the query. For each mismatch, the similarity matrix $S$ is used to identify the non-query term in the recipe that is most
similar to each absent query term. The recipe text to be reused is rewritten with
the new ingredient replacing every occurrence of the old one. The replacement is
done for only ingredients and not for cuisines, ingredient types nor meal types.

4.3 Queries with Negation

These queries occur in two different ways: (1) as an ingredient that is undesirable;
or (2) as a dietary practice (e.g. nut-free). For (1) the query has 1 for desired
ingredients, -1 for undesirable ingredients, and 0 otherwise. The substitution
adaptation described above is repeated but this time ingredients that correspond
to any -1s in the query are substituted.

Negative dietary practices are handled differently. The query vector must
be created to represent the practice; e.g. setting the nut ingredient entry to -1.
The query can now be handled in the same way as queries of type (1). This
technique is also used to incorporate ad hoc rules which improve the retrieval
for meal types such as beverage, cake, candy, dessert, ice cream by setting the
meat and fish ingredient entries to -1.

4.4 Three-Course Menu

A query for a three-course menu is handled by explicitly specifying the meal type
for the first (appetizer, salad or soup) and third course (cake, candy, dessert, ice
cream, snack and sweet ) in addition to the ingredients. A three-step retrieval is
carried out by generating three queries. Each query contains the list of desired
ingredients and 2 meal types penalised. For example consider a query for soup,
dessert, with desired ingredients tomato, garlic, potato and orange Here the
system would negate terms main course and dessert when querying for the
first course; likewise negate soup and dessert when querying for main course;
and negate soup and main course when querying for dessert. The final system
recommendation is simply a combination of the best matches from each of the
three retrievals.

5 Using RaGoÜt

The software is delivered as ragout.jar. The ragout.zip³ contains the executable
jar and a readme text file. Figure 6 shows RaGoÜt’s user interface. The upper
section contains the requirements entry panels, and the lower section displays
retrieved recipes. The user transfers ingredients from the ingredient picker to the
desired or undesired lists, or may remove them from the selected lists. The user
can also select from available dietary practices, cuisine types and/or meal types
on the right. The results pane is displayed in two columns. The TI(tles) of the
sorted list of retrieved recipes appear at the left, from which the user can select
a particular recipe to display on the right.

³ http://www.comp.rgu.ac.uk/staff/iaa/ragout.zip
6 Initial Experiments and Conclusions

RaGoUT addresses all outlined CCC tasks: compulsory, negation and menu challenges. Retrieval is achieved by combining similarity computations from exact matching with a WordNet-based semantic similarity matching. This combination ensures any mismatches are well placed for substitution adaptation.

On average 4 out of RaGoUT’s 5 top recommendations for each of the compulsory tasks are very relevant. In particular with query 4 (turkey, pistachio and pasta) results in the retrieval of a recipe with chicken, pistachio and rice noodles with appropriate substitutions for chicken. Here noodles are deemed similar to pasta due to their relative closeness in the WordNet hierarchy. Similarly a generalised term such as meat in query 1, resulted in the retrieval of recipes containing either turkey, beef or chicken. However some substitutions are questionable such as when cauliflower is substituted for onions in query 1. When queried for a Chinese dessert with fruits (in query 3), RaGoUT’s recommendation included a lychee sherbet which is a typical Chinese fruit dessert. This is possibly due to the association between lychee and Chinese cuisine.
Results from the single negation query are also very satisfactory, with the top 3 being very relevant. However the 4th recipe *cheese and macaroni* is less relevant because, although it does not contain *garlic* or *cucumber*, it is not a salad. RaGoUt substituted *garlic* in the 5th recipe with *clove* making it relevant as it already contains *tomato* and *salad*.

RaGoUt’s approach to the three-course meal challenge is very much at an initial stage and needs further development. With this task, substitution in particular, remains difficult. We plan to address this by extending RaGoUt’s representation such that ingredients are differentiated from non-ingredients and weights are incorporated in the retrieval to allow the reuse of multiple recipes.

The cooking contest has allowed us to integrate our TCBR expertise and adaptation knowledge learning methods in the RaGoUt system. In particular, we have proposed a similarity-aware case authoring mechanism with minimal domain-specific knowledge. This knowledge-rich representation has allowed simple retrieval and similarity-focused substitution adaptation. The weighted combination of case similarity scores enables the system to address the trade-off that exists between cases containing matching ingredients with those that have similar alternative.

References