2008-01-01

Back to the Future:Knowledge Light Case Base Cookery

Qian Zhang
Technological University Dublin

Rong Hu
Technological University Dublin

Brian Mac Namee
Technological University Dublin, brian.macnamee@tudublin.ie

See next page for additional authors

Recommended Citation

Zhang, Q., Hu, R. & MacNamee, B. (2008). Back to the Future:Knowledge Light Case Base Cookery. In M. Schaaf (ed), *Proceedings of the Workshop on the Computer Cooking Contest, 9th. European Conference on Case-Based Reasoning*,,pg. .239-248. doi:10.21427/1hkq-vf87

Funder: Science Foundation Ireland under Grant No. 07/RFP/CMSF718.

Authors
Qian Zhang, Rong Hu, Brian Mac Namee, and Sarah Jane Delany

Back to the Future: Knowledge Light Case Base Cookery

Qian Zhang, Rong Hu, Brian Mac Namee, and Sarah Jane Delany

Dublin Institute of Technology, Dublin, Ireland
qian.zhang@student.dit.ie, rong.hu@dit.ie,
brian.macnamee@dit.ie,sarahjane.delany@dit.ie

Abstract. The domain of cookery has been of interest for Case-Based Reasoning (CBR) research for many years since the CHEF case-based planning system in the mid 1980s. This paper returns to look at this domain, emphasising a knowledge-light approach. Our approach focuses on; the design of a structured case representation which encapsulates the details of a recipe, on leveraging WordNet for identifying food items and the relationships between them, and on using Active Learning to assist in labelling recipes with meal and cuisine types. Users can search for recipes by specifying the ingredients they wish to include in, or exclude from, the recipe and optionally specifying the type of meal and/or cuisine they are interested in. Recipes are retrieved based on a weighted similarity of the ingredients, the meal and/or cuisine types (if specified) and the textual similarity between the query and specific fields of the recipe text. The system includes substitution adaptation where a recipe can be recommended with a replacement ingredient, where appropriate.

1 Introduction

Choosing and designing recipes is an attractive problem for CBR research. It is an accessible and well understood domain with many individuals believing themselves to be experts in the area. It has proven popular with CBR researchers over the years with CHEF [2] and Julia [3], being two of the more well-known CBR cookery systems. CHEF is a case-based planner which represents recipes as cases using goals and problems, while Julia is a case-based design system. Both systems include an extensive semantic memory to hold the definitions of the terms needed, using frames organised into a semantic network in CHEF, and a large taxonomy of concepts in Julia.

This paper addresses the first two challenges in the Casebase Cookery Challenge by designing a case-based cookery system called *What's in the Fridge?* (WitF). WitF offers a new approach to CBR cookery, a knowledge-light approach, where we attempt to minimise the domain knowledge incorporated into the system and leverage existing technologies where possible. Our approach focuses on three strands:

(i) the design of a structured case representation which encapsulates the recipe details,

(ii) leveraging WordNet to identify recipe ingredients and to measure the similarity between ingredients,
(iii) using Active Learning to label the recipes with minimal user involvement.

One of the main objectives of our approach to case-based cookery is to remove the need to build a domain-specific semantic memory to define the types of foods and the relationships between them. To do this, we have leveraged WordNet [1], an electronic lexical database, which provides us with an ontology of food and food-associated entities. WordNet provides facilities for us to identify the food product from the ingredient description and also to provide a measure of similarity (using path distances) between food products. This similarity is used when calculating the overall similarity between recipes and queries. WordNet also allows us to provide straightforward substitution adaptation [9] in recipes, where ingredients can be replaced by other ingredients which are measured as most similar based on path distance similarity.

Another key objective of our approach is to significantly reduce the manual annotation and labelling of the recipes. To do this we have used Active Learning [6]. Active Learning is a supervised learning approach that aims to construct accurate classifiers while minimising the labelling effort required from experts. An outcome of active learning is the ability to label large pools of unlabelled data with minimal user input. We have used Active Learning to label the recipes, categorising them as a type of dish (such as *Main Dish*, *Starter* and *Dessert*) and a type of cuisine (such as *Asian* and *Mediterranean*).

The system was developed in Java and used the data-binding framework Castor[1] to manipulate the XML documents. Java WordNet Library (JWNL) V1.3[2] was used to access the WordNet 2.0[3] dictionaries. JWordNetSim V1.0.0[4] was used for measuring similarity between concepts in WordNet. The system was deployed using an Apache Tomcat web server and is available online[5].

The rest of this paper is organised as follows. Section 2 describes the structured case representation used by the system. It describes how the original text was converted automatically to the new case representation and how active learning was used to add labels to the recipes in the case base. Section 3 discusses retrieval in the system and how similarity between the query and the recipes in the case base was measured. Section 4 discusses reuse in the system with conclusions and potential future work outlined in Section 5.

2 Case Representation

The recipes for the challenge were provided in a semi-structured textual format in an XML file. Each recipe consists of three parts; a title, a list of ingredients and

1 http://www.castor.org/
2 http://sourceforge.net/projects/jwordnet
3 http://wordnet.princeton.edu/
4 http://nlp.shef.ac.uk/result/software.html
5 http://whatfridge.dmc.dit.ie/witf/newsearch

a cooking process. All fields are text based. Fig 1 shows the XML representation of the original data.

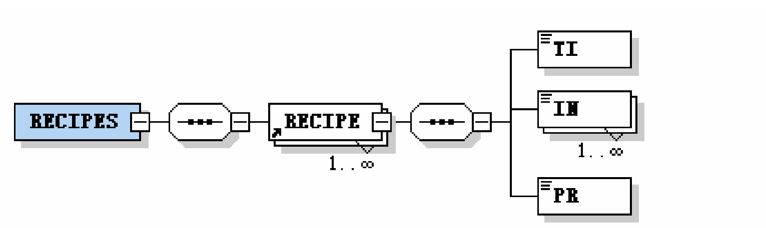

Fig. 1. XML representation of the original challenge data

The case representation used in WitF is displayed in Fig 2 as an XML schema. The case includes the original text fields of title (TI) and the cooking process (PR), but has replaced the list of text-based ingredients with a structured ingredient representation which is displayed in Fig 3. The representation of an ingredient includes a description, which contains the textual description from the original recipe structure, and a list of the actual food products that the ingredient includes. There can be more than one food product per ingredient as certain recipes include a list of optional products in a single textual ingredient description, such as $\frac{1}{4}$ *cup chopped walnuts or pecans*.

WordNet is used to identify the food products from the ingredient text. Each food product stored for an ingredient is a food concept from WordNet. The text is parsed, primarily using regular expressions, and the potential food products we identify are checked to ensure that each one is a node in one of a number of food related sub-hierarchies in WordNet. The sub-trees used were senses 1 and 2 of the noun *food* to include most food stuffs; sense 1 of the noun *fruit* to include fruit and nuts; sense 1 of the noun *edible fats* to include oil, lard etc. and sense 1 of the noun *leaven* to include yeast, dough etc. Using this process we successfully identified 570 individual food products and were left with approx 80 unidentified food products, mainly due to spelling mistakes or unusual items not listed in WordNet (e.g. prosciutto[6] or jicama[7]).

We also identify and associate a level with each ingredient in the recipe. A level 1 ingredient is considered to be the main, or one of the main ingredients, in the recipe. A level 2 ingredient is an important but non-main ingredient in the recipe and a level 3 ingredient is considered to be a flavouring or seasoning ingredient. The levels of ingredients are identified by the quantity associated with each ingredient and are set by comparing all ingredients in a single recipe.

The information stored for each valid food product includes:

[6] A dry-cured ham from the Italian word for ham
[7] An edible tuber known as the Mexican potato or turnip

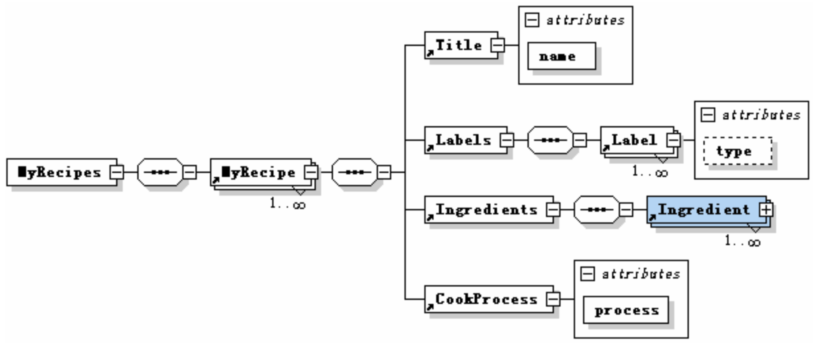

Fig. 2. XML representation of the case

(i) name - the base form of the noun in WordNet (known as the *lemma*) that is a node in a food-related sub-tree.
(ii) sense index - the meaning of the word in WordNet. Each sense of a word is associated with a different synset or set of synonyms of the word.
(iii) quantity - the quantity of the food product needed for this ingredient in the recipe, e.g. $\frac{1}{2}$ in $\frac{1}{2}$ *c of pecans; chopped*.
(iv) unit - the unit measurement related to the quantity, e.g. *cup* in $\frac{1}{2}$ *c of pecans; chopped*.
(v) format - any specific formatting instructions associated with the ingredient, e.g. *chopped* in $\frac{1}{2}$ *c of pecans; chopped*.
(vi) isOptional - a boolean value indicating whether the food product is optional or not, e.g. pecans are not optional in $\frac{1}{2}$ *c of pecans; chopped* but are optional in $\frac{1}{2}$ *c of pecans or walnuts; chopped*.
(vii) level - the level of the ingredient, either 1, 2 or 3.

In addition a list of text-based labels is added to the case, representing the type of course and/or the type of cuisine by which the recipe can be categorised. The details of how the labels are added to each recipe are covered in Section 2.1 below.

2.1 Labelling using Active Learning

Active Learning is a supervised learning approach that aims to construct accurate classifiers using a minimal number of labelled examples. The purpose of this is to minimise the labelling effort required from human experts. Active learning incrementally builds a classifier by iteratively selecting the most informative example from an available pool of unlabelled cases, presenting this to to the expert for labelling, incorporating the labelled case into the classifier's training data, and rebuilding the classifier [6].

Another use for active learning is as a tool for labelling large collections of unlabelled data. There have been some examples of this particularly in image [4]

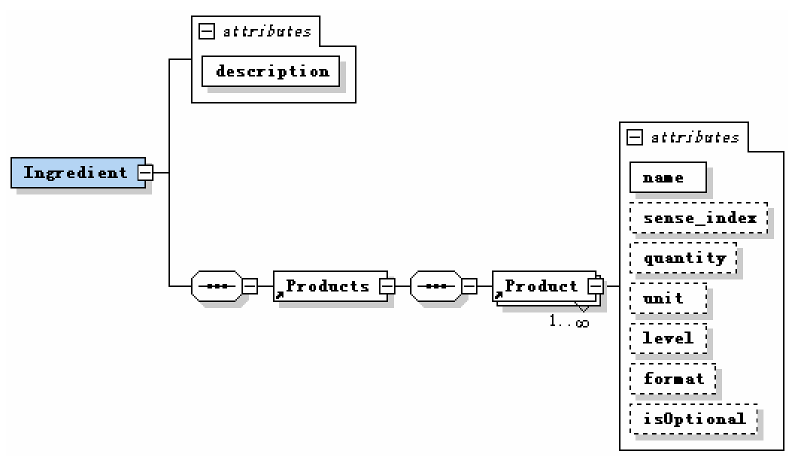

Fig. 3. Representation of the ingredient in a case

and video [8] labelling. For WitF an active learning system was built to label each recipe in the challenge dataset to indicate for which courses it is appropriate, and under which cuisine types it falls. Each recipe can be suitable for a number of different courses and, less frequently, a number of different cuisine types. As each case can have a number of labels applied to it a series of two-class active-learning-based labellers are built, each of which labels recipes as belonging to a target class or the class of *others* (e.g. desserts or others, Asian or others, etc). Our goal is that the entire recipe case base can be labelled with a large number of labels without a significant amount of manual labelling.

The active learning algorithm used is a relatively simple one, and only deals with two-class problems. The algorithm starts with a large pool of unlabelled cases each of which is to be labelled as belonging to one of two classes. A small number of cases of each class are initially selected (presently at random) from this pool and are manually labelled by the expert. These labelled cases are used to build a ranking k-nearest neighbour (k-NN) classifier which uses distance-weighted voting [7]. The algorithm then proceeds by using this classifier to classify each case in the remaining pool of unlabelled cases. The classifier returns for each case c the value *posScore(c)* which is the sum of the similarities between the query case c and any nearest neighbours belonging to the positive class and *negScore(c)*, the sum of the similarities between c and any nearest neighbours belonging to the negative class. These values are normalised as in Equation 1 and the pool is ranked according to these scores.

$$normPosScore(c) = \frac{posScore(c)}{posScore(c) + negScore(c)} \tag{1}$$

The case in the pool with a *normPosScore* nearest to 0.5 is removed from the pool and selected as the next case for labelling by the expert. This is the case that the system is currently most uncertain about and so labelling it will result in the most benefit to the system. After the case is labelled it is added into the case base and the pool is re-labelled and re-ranked. This process continues until a specified number of labels (the *label budget*) have been manually added by the expert. A flow diagram of this process is shown in figure 4. The retrieval process and the similarity measures used in the active learning classifier are similar to those described in section 3.

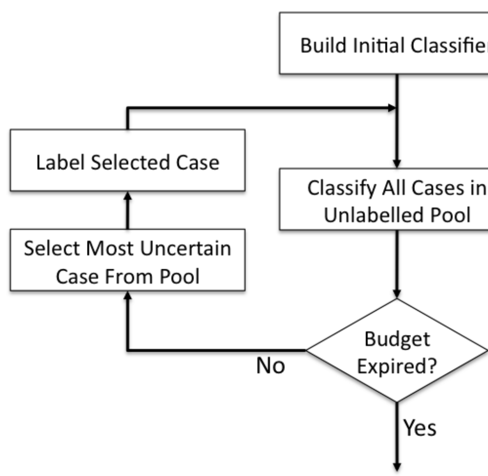

Fig. 4. The active-learning-based labelling process.

To estimate the accuracy of the labelling system, an expert labelled each recipe in the case base indicating whether or not it was a *dessert*. The active learning process was then used to label each recipe in the case base also as dessert or not dessert. These labels were compared to the expert's labels and an accuracy figure was calculated. In this experiment labelling was allowed to continue to a budget of 400 labels in order to see how accuracy improved with label count. Fig 5 shows the result. As can be seen from the graph 90% labelling accuracy is achieved after just 80 manual labels - less than 10% of the pool. Further analysis on the other eight labelling tasks showed that this 10% figure represented a good compromise between accuracy and labelling burden.

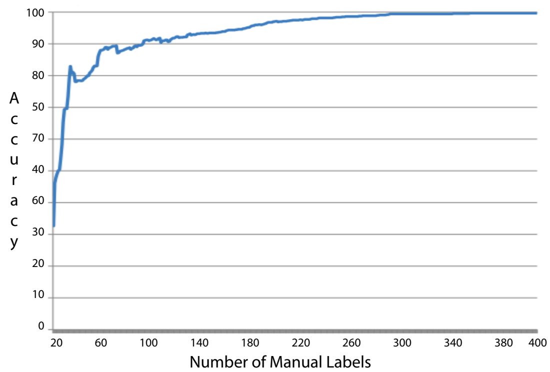

Fig. 5. Labelling accuracy compared to the number of manual labels.

3 Retrieval

WitF can be queried by entering, in free-text format, a list of ingredients that should be included or not included in any recommended recipe, and by optionally selecting a specific type of course and/or cuisine. Each ingredient entered is validated as a food concept in WordNet, in the same way as the identification of the ingredient food products in a recipe case described in Section 2.

The system retrieves the k recipes[8] from the case base that best match the query. The measure used for comparing each recipe with the query is a weighted measure of the similarity between the query and recipe ingredients, the textual similarity of the title of the case base recipe to the query ingredients and the similarity between any specified types. This is shown in Equation 2 where r is a case base recipe, q is the query, W_i are weights, r_{title} is the case base recipe title, r_t are valid recipe types, q_t are any types specified in the query and $\delta_{r_t q_t}$ is Kronecker's delta where $\delta_{ij} = 1$ if $i = j$ and 0 otherwise.

$$Sim(r,q) = W_{ing}Sim_{ing}(r,q) + W_{title}cosine(r_{title},q) + \sum_{r_t \in r}\sum_{q_t \in q}\delta_{r_t q_t}W_{type} \quad (2)$$

When calculating the ingredient similarity between a case base recipe r and query q, the recipe ingredient in recipe r that best matches query ingredient q_{ing} is found for each query ingredient. This is known as r_{best} for recipe r. r_{best} is selected as that ingredient in the recipe which has the highest similarity to the specified query ingredient, q_{ing}.

The overall ingredient similarity for recipe r and query q is then calculated as a weighted sum of the similarities between each query ingredient and its appropriate best match ingredient in the recipe as given in Equation 3 where

[8] Currently $k = 5$ but this is configurable

r_{best} is the best match ingredient in recipe r for query ingredient q_{ing} and $W_{r_{best}}$ are recipe ingredient weights.

$$Sim_{ing}(r,q) = \sum_{q_{ing}} W_{r_{best}} Sim(r_{best}, q_{ing}) \qquad (3)$$

The individual recipe ingredient weights, $W_{r_{best}}$ are related to the level of the r_{best} ingredient in the recipe as identified in the case representation, see Fig 2. This allows the main ingredients in a recipe to have more of an influence in the calculation of similarity than non-main ingredients. In fact, level 3 ingredients are considered unimportant from a similarity point of view (rather than a taste point of view) and are not included in the calculation of similarity for the recipe i.e. $W_{r_{best}} = 0$ where the level of r_{best} is 3.

The measure used for calculating the similarity between two ingredients is Jiang and Conrath's measure [5]. This measure is described in Equation 4 where ing_1 and ing_2 are the food concepts, IC is the information content of a concept[9] and LCS is the least common subsumer, i.e. the most specific concept description that includes ing_1 and ing_2. Each ingredient has previously been validated as a concept in the WordNet ontology and this measure uses the shortest path length between two concepts in an ontology and the density of the concepts along this path to calculate similarity between two concepts. The similarity scores are normalised to fall in the range of 0–1.

$$Sim(ing_1, ing_2) = \frac{1}{IC(ing_1) + IC(ing_2) - 2 \times IC(LCS)} \qquad (4)$$

The textual similarity $cosine(r_{title}, q)$ used in Equation 2 is calculated as the cosine similarity between the recipe title and a concatenated string of all the included query ingredients where the term weights of the terms in both strings are normalised to unit length for each string. However, in the case of the active learning labelling process the textual similarity compares two recipe titles to each other as the active learning classifier will always be comparing complete existing recipes to each other, and so full text titles will always exist.

3.1 Excluding ingredients in retrieval

The user can specify ingredients that they want to exclude from any recommended recipes. Using WordNet provides us with a very straightforward mechanism for ensuring no recipe which includes any of these ingredients is retrieved. Firstly, the sub-tree of the food sense (i.e. the noun and sense index) of the excluded ingredient is retrieved from WordNet. Only recipes that do not include any ingredient that is a node in the sub-tree will be considered for retrieval. For example, if the user is vegetarian they simply need to include *meat* as the excluded ingredient and any recipe with an ingredient that is a node on a sub-tree of *meat*, such as *chicken* or *pork* will be excluded from retrieval.

[9] The conventional way of measuring IC is to combine hierarchy information of an ontology with statistics on their actual usage in a large text corpus. $IC(c) = log^{-1}P(c)$ where $P(c)$ is the probability of occurrence of the concept c.

4 Reuse

The similarity measure used in WitF provides for a partial matching situation, not requiring the retrieved recipes to exactly match the query ingredients. The identification of r_{best} facilitates this by identifying the most similar ingredient in the recipe to the query ingredient. In effect, substitutional adaptation is a by-product of this similarity measure. Recipes with ingredients not the same as the query ingredient but related or 'similar' to the query ingredient can be retrieved within the top k recipes. For example, the query of *meat* shown in Fig 6 returned *turkey* as the best match ingredient in the highest recommended recipe.

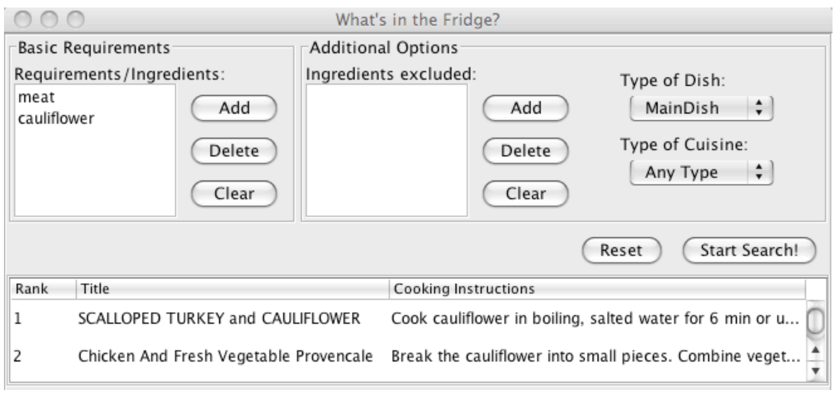

Fig. 6. *What's in the Fridge?* Screenshot

The retrieval algorithm uses a threshold to ensure the relevance of the retrieved recipes. The recipe threshold score $thres_{recipe}$ is the recipe similarity score below which a recipe will not be returned to the user, even if it is within k. In addition, a recipe that has an individual ingredient similarity score less than the ingredient threshold score $thres_{ing}$, i.e. $Sim(r_{best}, q_{ing}) < thres_{ing}$ will not be retrieved. The reason for this is that the best ingredient match is too poor a match to allow for substitution.

The fact that any valid WordNet food concept can be entered as a requirement in the query allows for some generalisation in the querying. Consider a query for a nut-free cake. This can be entered by selecting the *Dessert* category of dish type, by entering *nut* as an excluded ingredient and by entering *cake* as a requirement. WitF returns a single recipe for this query due to the thresholding described above; the recipe returned is *Berry Chocolate Pie*. This technique will also allow specifying certain requirements such as *salad* or *soup*. So searching for a recipe for *eggplant soup* can be specified as two requirements; *eggplant* and *soup*. Since there is no recipe for *eggplant soup* in the dataset this query returns five various vegetable soup recipes, where the system has partially matched the *eggplant* ingredient with other vegetables.

5 Conclusions

In this paper we have described our knowledge-light case-based cookery entry for the 1st Computer Cooking Contest. The objectives of our system were to minimise the domain knowledge incorporated into the system and leverage existing technologies. Due to its knowledge-light design, our system offers a number of compelling advantages in terms of scalability and maintainability. Adding new recipes to the system simply involves converting them to our case representation, which is an automatic process involving no user input, and using the active learning process to label them. This will request the user to label only 10% of the additional recipes with each label. In addition the system can be easily extended to add new dish or cuisine types. This involves selecting a number of recipes to act as the initial training data for the active learner and then labelling the recipes presented by the active learner.

Acknowledgements

We would like to thank Evin McCarthy, John McAuley, Paula McGloin and Anna Deegan for their help in the implementation of the web version of WitF.

This material is based upon works supported by the Science Foundation Ireland under Grant No. 07/RFP/CMSF718.

References

1. Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database*. MIT Press, 1998.
2. K. J. Hammond. Chef: A model of case-based planning. In *AAAI Proc of the 5th National Conf on Artificial Intelligence*, pages 267–271. Morgan Kaufmann, 1986.
3. T. R. Hinrichs. Strategies for adaptation and recovery in a design problem solver. In Hammond, editor, *Proc Second Workshop on Case-Based Reasoning*. Morgan Kaufmann, 1989.
4. S.C.H. Hoi and M.R. Lyu. A semi-supervised active learning framework for image retrieval. *Computer Vision and Pattern Recognition*, 2(20-25):302–309, 2005.
5. J. J. Jiang and D. W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proc of ROCLING International Conference on Research in Computational Linguistics*, 1997.
6. D Lewis and W Gale. A sequential algorithm for training text classifiers. In *Seventeenth Int ACM-SIGIR Conf on Research and Development in Information Retrieval*, pages 3–12. Springer Verlag, 1994.
7. T. Mitchell. *Machine Learning*. McGraw Hill, New York, 1997.
8. J. Yang R. Yan and A. Hauptmann. Automatically labeling video data using multi-class active learning. In *Ninth Int. Conf on Computer Vision (IICV '03)*, pages 516–523, 2003.
9. W. Wilke and R. Bergmann. Techniques and knowledge used for adaptation during case-based problem solving. In *IEA/AIE (Vol. 2)*, pages 497–506, 1998.