JaDaCook 2: Cooking Over Ontological Knowledge

Conference Paper · January 2009

CITATIONS
6

READS
82

4 authors, including:

Pedro Javier Herrera
National Distance Education University
46 PUBLICATIONS **252** CITATIONS

SEE PROFILE

JaDaCook 2: Cooking Over Ontological Knowledge

P. Javier Herrera, Pablo Iglesias,
Ana Mª García Sánchez, Belén Díaz-Agudo

Department of Software Engineering and Artificial Intelligence,
Universidad Complutense de Madrid, Spain.
{pjherrera@pdi.ucm.es, belend@sip.ucm.es}

Abstract. JaDaCook 2 has been developed to participate in the 2nd *Computer Cooking Contest* 2009 (CCC-09). The system is an improved version of JaDaCook 1.0 that participated in CCC-08. We have reengineered the source code making the code more reusable and extensible. JaDaCook 2 includes general improvements and new functionality. Namely, a new form based interface, a new version of the ontology including more type of ingredients, dietary practices, types of meal type of cuisine, data mining over the ingredients, textual and IE capabilities over the recipes. The system has been developed as the final evaluation assignment for a graduate course in Machine Learning at the Computer Science Faculty (*Complutense University of Madrid*). In this paper we present a brief review of the technical characteristics of the system, describing the knowledge acquisition and reasoning processes, the new functionality and some experimental results.

Keywords: Knowledge Intensive CBR, Ontology, jCOLIBRI 2, Data Mining

1 Introduction

In this paper we describe JaDaCook[1] version 2, a CBR system that solves the task of suggesting a recipe given a restricted set of ingredients as the query. This version of the system has been developed to participate in the Computer Cooking Contest 2009. It is an improved version of the JaDaCook 1 that participated in CCC-08.

The system has been developed by students as the final evaluation assignment for a graduate course in Machine Learning at the *Computer Science Faculty (Complutense University of Madrid)* during the first semester of 2009. They have reengineering the source code of JaDaCook 1 making the code more reusable and extensible. JaDaCook 2 includes general improvements and new functionality. Namely, a new form based interface, a new version of the ontology including more type of ingredients, dietary practices, types of meal, type of cuisine, data mining over the ingredients, textual and IE capabilities to process the text of the given recipes.

JaDaCook 2 reasons using different knowledge sources: (1) a case base of recipes (provided by the CCC-09 organizers and available from textual sources), (2) a cooking ontology, (3) a set of association rules, obtained using data mining techniques, capturing co-occurrences of ingredients in the recipes. These rules are

[1] http://gaia.fdi.ucm.es/grupo/projects/cookingContest/cookingContest.html#jadacook2

used to propose substitute ingredients, (4) a case base of menus built collaboratively using the personal opinion of non-expert users. This case base is used to build menus by the composition of single dishes.

JaDaCook 2 solves the *Compulsory Task* of the CCC-09 that involves answering queries that require the selection and, where appropriate, modification of a recipe for a single dish; and the *Menu Challenge* that requires the composition of a three-course menu based on the available recipes and on collaborative recommending techniques.

JaDaCook 2 adapts the retrieved recipe by substituting ingredients; however it does not aim to solve the *Adaptation Challenge* in CCC-09 as it does not change the preparation directions in a recipe.

JaDaCook 2 has been implemented using the jCOLIBRI [1] framework. jCOLIBRI helps the development as it supports, among other things, textual processing [2] and the use of ontologies to enrich reasoning processes in CBR systems [3].

JaDaCook 2 includes a new form based interface based on the multiplatform framework QT [7]. The new interface replaces the previous natural language interface, simplifies the query formulation process, and minimizes errors in the communication process with the user. The new interface allows navigating the case base, querying it by selecting from the ontology specific ingredients or types to be included or avoided and dietary practices. It also allows querying the system using new ingredients, including new ingredients in the ontology, new menus in the collaborative case base and new recipes that are automatically processed from the given XML file.

The Menu Challenge asks for the creation of a three course menu. JaDaCook 2 offers a case based collaborative recommender system that bases the menu configuration on the opinion of previous users. When the user queries the menu system, (s)he is asked about her opinion on the result, and his/her answer is recorded and used for future recommendation.

In this paper we present a brief overview of the technical characteristics of the system, describing the knowledge acquisition and reasoning processes, the new functionality and some experimental results with the queries provided in the CCC-09 webpage.

The paper is structured as follows. Section 2 details the main features of the graphical interface of the system. Section 3 describes the knowledge sources of the system and Section 4 briefly explains the CBR processes. Section 5 offers some examples and results and Section 6 concludes the paper.

2 Graphical Interface

The new interface of JaDaCook 2 integrates in the same GUI (see Fig. 1) three tabs (from left to right in Fig.1) that correspond to the following tasks:

1) **Single Dish Challenge**. Given a query, the result will be the retrieval and adaptation of a recipe for a single dish.

2) **Menu Challenge**. This function enables the creation of a three course menu. The collaborative part uses the opinions of previous users to guide future recommendations.

3) **Recipes inspection** where the user can consult the case base.

Next subsections describe these three features of the system.

Fig. 1. JaDaCook Main Window: single dish tab

2.1 Single Dish Tab

In the single dish tab (Fig. 1) the user provides the query information. It is not mandatory to include every component:

- List (WL) of Ingredients that the user would like to include.
- List (WNL) of ingredients that the user would not like to include.
- Dietary practices like Vegetarian, Nut-free or non-alcoholic, plus new dietary practices like gout diet or cholesterol diet.

- Type of cuisine. One of the following: Chinese, Mexican, Mediterranean or Italian.

Selecting corresponding radio button on left and right sides of single dish tab (see Fig. 1) the user can introduce one by one the list of ingredients that the user "would like to" (WL) and "Wouldn't like to" (WNL) include in the recipe, respectively.

There are two ways to add an ingredient to the query lists WL and WNL of ingredients:

1) Select an ingredient from the ontology. If this button is clicked then a new window shows the ontology tree where the user can select the ingredient, either specific, like orange or salmon, or generic, like Fruit or Fish.
2) Write the name of the ingredient in the text box provided to it. If the word entered by the user is a new ingredient, then the user is given the option to add it to the ontology of ingredients for future reference or use instead an ingredient existing in the ontology. A new ingredient entered by the user is ignored in the current query.

The *Suggest ingredient* button can be used to let the system complete the query WL list with other ingredients that are compatible with the ones previously included by the user. These suggested ingredients are obtained from the recipes case base, and are ingredients that typically appear together. This utility is one of the novel features of the new version of the system and it is described in Section 3.3.

The user can remove an ingredient from the query "I would like" or "I wouldn't like" lists with the *Remove* button after selecting the ingredient.

Once the query data have been introduced the system computes similarity and filters the k more similar recipes for the given query. The k value is configurable through the file menu. The resulting recipes are shown and annotated with a label that explains why this recipe has been retrieved. (R) annotates an ingredient of the WL list that appears in the retrieved recipe. (A) annotates an ingredient that is in the WL list but not in the recipe, so it has been substituted by a similar ingredient. And (C) annotates a specific ingredient that appears the WL list.

2.2. Menus Tab

The menus tab allows the user to compose three courses menus. The user starts by using one of the buttons "Starter" "Main Course" or "Desert" to query the system for a single dish. The system uses the same interface described in the previous section. Once the user enters a query for a starter, main course or dessert then the recommender system makes a secondary search to complete the rest of the menu.

The recommender subsystem has a case base of menus that is initialized with a file named "*menus.xml*". The menu recommender system searches in the menu case base for menus that are similar, i.e., they have similar courses, to the query menu, that is partially described menu. High scored menus in the case base are selected. Menu case base has been built collaboratively using the personal scores of non-expert users. The

"punctuation" button let you associate good/bad score to the menu. This score will be used in future recommendation cycles.

In Fig. 2 we see an example. The user queries for a starter with chicken and Chinese as the cuisine type. With the first choice "Asian Chicken Salad W/spicy Peanut Sauce", the system in the menus tab recommends a main course and a dessert.

Fig. 2. JaDaCook Main Window:menus tab

2.3 Recipes Tab

This tab offers a simple inspection utility of the 1484 recipes in the case base. The recipes are organized in 149 pages where the user can easily look for specific recipes.

3 Knowledge Acquisition

In JaDaCook 2, the knowledge of the system has evolved. Apart from the case base, the main source of knowledge of JaDaCook 2 is the ontology where we have conceptualized and formalized cooking knowledge from different sources, including experts, dictionaries and cooking web pages and systems [4-5]. In addition, the CCC-09 organizers provided a large case base of recipes for use by contest participants.

The acquired knowledge has been incrementally structured, conceptualized and formalized in the Web Ontology Language OWL[2] and the new version includes ingredients from the new recipes, new types of ingredients, new types of cuisine and dietary practices, and classification of recipes and ingredients according to these dietary practices, namely gout diet knowledge[3], seasonal food knowledge[4], and cholesterol diet knowledge[5].

3.1 Ontology

Like its predecessor JaDaCook 2 is based on an ontology that captures the terminological knowledge of the cooking domain, organizing objects (individuals) into categories (concepts) to enable inheritance of properties and to create taxonomies. There are animal origin ingredients, grouped as fish, meat, milk, cheese and eggs, each with other subclasess; plant origin ingredients, like cereals, nuts, fruits and vegetables; and other classes like sweeteners, drinks, and basic ingredients like salt and oil. In the new version of the ontology we have included additional knowledge. It has more than 300 ingredients, organized in types and classes that cover the 1484 recipes provided by the CCC-09 organizers. The OWL code of the ontology is available through the web page:
http://gaia.fdi.ucm.es/grupo/projects/cookingContest2/jadacook2/OntologiaIngredientes.owl

3.2 Cases

In the new version of the system we use the case base provided by the CCC-09 organizers. Cases are stored in an XML file, that is processed and the most relevant information is extracted and stored in case base memory structure in the precycle of the application. Processes of loading and storing cases have also been improved in this version using SAX and DOM [8]. Each case in the case base has mainly a text title, a list of ingredients and a textual description of the recipe development process. Each case is linked with the corresponding classes of ingredients in the ontology.

---

[2] http://www.w3.org/2004/OWL/
[3] Gout diet knowledge: http://www.everydiet.org/diet/gout-diet
[4] Seasonal food knowledge:
http://www.comidacasera.com/especiales/dietas/alimentos_temporada.phtml
[5] Cholesterol Diet knowledge: http://gicare.com/Diets/low-cholesterol-diet.aspx

3.3 Dependency Rules between Ingredients

We have applied the Apriori algorithm [9] to mine association rules using WEKA over the case base of recipes. The data set is a csv file obtained from the recipes case base, so it includes 1484 rows, and one Boolean attribute for each ingredient in the ontology. WEKA allows the resulting rules to be sorted according to different metrics such as confidence, leverage, and lift. We have performed different experiments with different values to extract a set of relevant rules. For example, the following rule indicates that if the recipe includes vanilla then it will include sugar with a confidence of 87% and support of 205 recipes.

vanilla=yes 205 ==> sugar=yes 179 conf:(0.87)

These rules allow capturing the degree of compatibility between ingredients. When the user asks for ingredients to include in the recipes, the system uses these rules to suggest ingredients that appear together with the previously selected ingredients in a large percentage of recipes.

3.4 Menu Case Base

The menu case base includes 133 menus acquired in a collaborative way by different users of the system. Each menu is composed of three courses taken from the recipe case base. If a user composes a menu using three single dish queries, then the menu is saved in an XML file to be reused in the future. Each menu has a numeric score that can be modified. This score represents the degree of satisfaction of the non-expert users that have tried this menu.

```
<MENU>
<PR>Fusilli Verde with Broccoli and Red Bell Pepper</PR>
<SE>Fast with Five: Garlic Flank Steak With Onion</SE>
<PO>Cocoa Espresso Cooler</PO>
<NOTA>7</NOTA> </MENU>
```

New menus and their scores will be included in the menu case base and will be taken into account for future menu recommendations.

4 CBR Processes

CBR processes in JaDaCook 2 take advantage of the ontology. The main usage of the cooking ontology has been centered on similarity assessment and ingredient substitution. The assumption here is that two ingredients are more similar if they are located closer in the ontology. And that one ingredient can be substituted for an ingredient that is classified in the same concept. For example, *turkey* and *chicken* are siblings and children of the concept “Fowl” (like *hen*, and *duck*).

The single dish case retrieval process consists of obtaining recipes that are similar to the given query. That means that the retrieved recipe includes the ingredients in the

WL list and it does not include ingredients from the WNL, has the dietary restrictions and type of cuisine specified in the query through the graphical interface (Fig. 1).

The retrieval method is k-nearest neighbor, which compares these characteristics and aggregates them obtaining a ranking similarity number to order the candidates. Then the k most similar are shown to the user. In JaDaCook 2 we have improved the similarity measure, and included more knowledge from the ontology.

Similarity assessment is based on two local similarity functions to compare titles and ingredients of the query and the case. Once all the ingredients have been processed, the overall similarity score is [0,1] normalized. Then the k most similar recipes will be shown to the user.

We use jCOLIBRI similarity functions, more specifically one of the concept based similarity functions that depends on the location of the cases in the ontology. Details about concept based similarity in jCOLIBRI can be found in [3]. If the ingredients in the WL list are concepts of the ontology that represent type of ingredients, then the children are obtained. To that end, we used the library OntoBridge [6] written in Java that provides management ontologies.

The reuse process in JaDaCook 2 is a process based on the substitution of ingredients according to their position in the ontology. The reuse strategy is the same one that was employed in JaDaCook 1 and it is called reinstantiation. It is one of the reuse methods included in the library of jCOLIBRI 2 and it is based on substituting one ingredient by one of its siblings in the ontology structure.

The stage of learning (retain) has as its primary function storing those cases which have been adapted into the knowledge base. Thus, these new cases may be used in future searches to improve retrieval performance. In this system, it is left to the user to decide which cases are retained, so in this way we store the cases that are most useful in problem solving.

5 Results and Examples

Next we provide the preliminary results for the queries in the single dish challenge provided by the CCC-09 organization through the web page. The rest of the queries, captures, the code of the system, and the documentation is available through the web page: http://gaia.fdi.ucm.es/projects/cookingContest2/cookingContest.html#JaDaCook

Q3: Prepare a low-cholesterol dessert with strawberries and avoid citrus fruits. (Main focus: dietary practice)

The WL list includes the 'strawberries' ingredient and we want to avoid 'citrus' so we include it in the WNL list. Dietary practices is set to 'Cholesterol diet'.

Fig. 3 shows the 3rd recipe suggested by the system, 'Fruit Cup#1' with a similarity value of 0.67. It includes strawberries and none of the citrus fruits: orange, lemon, lime and tangerine.

Q9: I do have a filet of beef, carrots, celery, field garlic and cucumber. Potatoes are available, too. For the dessert, we have oranges and mint. A soup would be preferable for the starter.

Q9 belongs to the Menu challenge. The WL list of the main course includes: 'beef', 'carrot', 'celery', 'garlic', 'cucumber' and 'potato'. The system answers with recipe 'Beef Stew with Zucchini' that includes 'beef', 'celery', and 'potato'. It also adapts the recipe

by replacing 'onion' by 'garlic', 'squash' by 'cucumber', and 'celery' by 'carrot'. For the dessert we choose 'orange' and 'mint' as required ingredients. The system answers with the recipe 'Indian Ice Tea'. As the starter the system does not allow looking for 'soup' as it is not available as an ingredient, or a type of ingredient, or dietary practice.

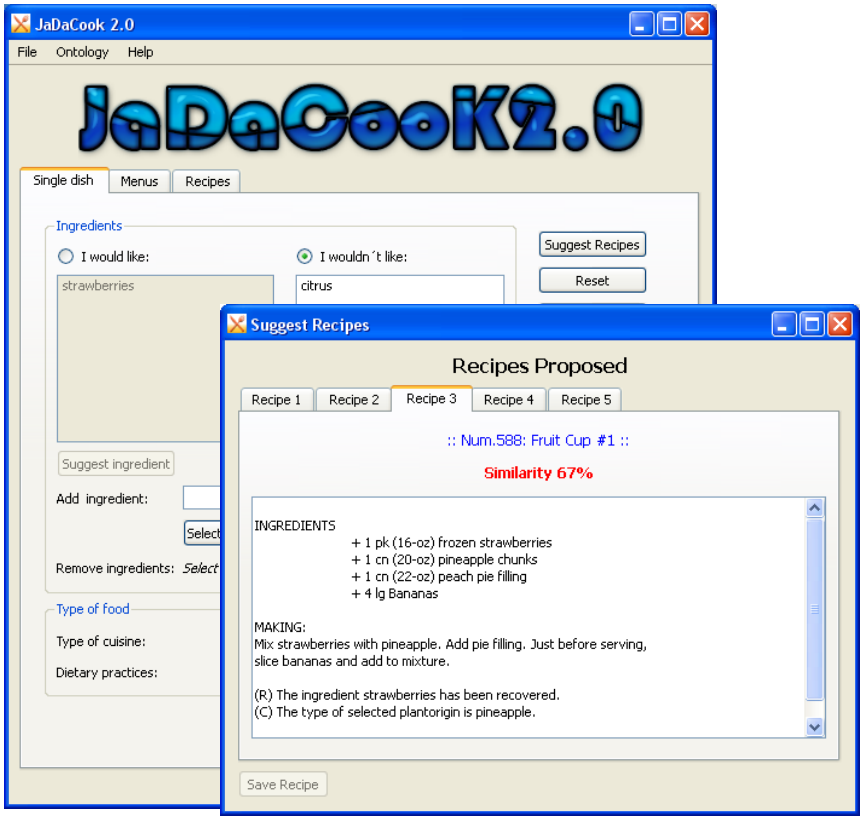

Fig. 3. Results for the query 3 of the single dish challenge provided by the CCC-09

6 Conclusions

In this paper we have described JaDaCook 2, a CBR recipe creation system submitted to the Computer Cooking Contest at ICCBR 2009.

JaDaCook 2 is a new version of JaDaCook 1 that participated in the 1st CCC in 2008. It addresses the main drawbacks of the first version, includes new functionality, improvements on the ontology, a new interface, and data mining capabilities to capture dependencies between ingredients.

JaDaCook 2 reasoning is based on a case base of recipes and an ontology with reusable knowledge about ingredients, types of ingredients, types of cuisine and dietary practices. The ontology is used as background knowledge to measure

similarity between ingredients and single dishes, and to substitute ingredients during adaptation.

The system is able to deal with the single dish and menu challenges, and it is able to learn new ingredients appearing in queries by including them into the hierarchical organization of ingredients. This is a kind of supervised learning. The system also learns new menus, as combinations of existing individual dishes.

The results for the given queries, the code of the system, and the documentation is available through the web page:
http://gaia.fdi.ucm.es/projects/cookingContest2/cookingContest.html#JaDaCook

Acknowledgments

This work has been supported by Spanish Ministry of Science and Education (TIN2006-15140-C03-02) and it has been partially supported by the Comunidad de Madrid Education Council and UCM(consolidated research group 910494).
We also thank the UCM students who participated in the development of JaDaCook 1: David Romero and Ignacio Rubio.

References

1. Díaz-Agudo, B., González-Calero, P.A., Recio-García, J.A., and Sánchez, A. Building CBR systems with jCOLIBRI. Special Issue on Experimental Software and Toolkits of the Journal Science of Computer Programming, (2007). Elsevier Science.
2. Recio-García, J.A., Díaz-Agudo, B., Gómez-Martín, M.A., and Wiratunga, N. Extending jCOLIBRI for Textual CBR. Proceedings of Case-Based Reasoning Research and Development, 6th International Conference on Case-Based Reasoning, ICCBR 2005, Chicago, IL, US, Springer, (2005) 421-435.
3. Recio-García, J.A., Díaz-Agudo, B., González-Calero, P.A., and Sánchez, A. Ontology based CBR with jCOLIBRI. In R. Ellis, T. Allen, and A. Tuson, editors, Procs. of AI-2006, the Twenty-sixth SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence, Springer. (2006) 149–162
4. Food Classification: http://www.mhlw.go.jp/english/topics/foodsafety/positivelist060228/dl/r04.pdf
5. Hammond, K. CHEF. In: Inside case-based reasoning. Hillsdale, NJ: Erlbaum.(1989).
6. OntoBridge: http://gaia.fdi.ucm.es/grupo/projects/ontobridge/
7. QT Qt Cross-Platform Whitepaper http://www.qtsoftware.com/products/
8. T. Cheung Lam, J. Jason Ding, J-Charn Liu. XML Document Parsing: Operational and Performance Characteristics. IEEE 2008.
9. Agrawal R, Imielinski T, Swami AN.:Mining Association Rules between Sets of Items in Large Databases.In *SIGMOD* **22**(2) (1993) 207-216