Evaluation of knowledge based applications: benchmark and guidelines

Sylvie Després
Laboratoire d’informatique médicale & Bioinformatique (LIM&Bio), Université Paris 13
UFR de Santé Médecine et Biologie humaine, 74, rue Marcel Cachin 93017 Bobigny France
sylvie.despres@univ-paris13.fr

Haïfa Zargayouna
Laboratoire d’Informatique de l’université Paris-Nord (LIPN) - UMR 7030
Université Paris 13 - CNRS
99, avenue Jean-Baptiste Clément - F-93430 Villetaneuse, France
haifa.zargayouna@lipn.univ-paris13.fr

Abstract—In this paper we propose to report a work on evaluation of a knowledge based application that leads to a constitution of a benchmark such as those that exist in Information Retrieval evaluations. This benchmark enables to perform quantitative evaluation by classic metrics such as precision and recall. We had also conducted a qualitative analysis that helps the elaboration of guidelines and methodological indications for ontology evaluation and enhancement.

I. INTRODUCTION

In recent years the number of knowledge based applications development has increased considerably. The need to evaluate the knowledge used in these applications is widely acknowledged. Unfortunately, experiences on evaluation has several drawbacks: (i) the observations made are difficult to generalize; (ii) the studies are barely reproducible; (iii) the results are mainly qualitative and are difficult to interpret: what is evaluated, the quality of the knowledge base or the way it is used?

In this paper we propose to report a work on evaluation of a knowledge based application that leads to a constitution of a benchmark inspired of test suites that exist in Information Retrieval (IR) evaluations. This benchmark enables to perform quantitative evaluation by classic metrics such as precision and recall. A quantitative evaluation is useful to get an overall idea of systems performances on shared resources. We had also conducted a qualitative analysis that leads to the elaboration of guidelines and methodological indications for ontology evaluation and enhancement.

The chosen application was the Computer Cooking Contest. We started the evaluation with resources of one of the participant systems (TAAABLE system). However the result can be useful for all participant systems. The next section presents in detail the context of this work.

II. CONTEXT

The TAAABLE system constitutes the framework of this study. Its aim is to answer to the Computer Cooking Contest challenge by using a Textual Case Based Reasoning system. Our task was to evaluate the quality of the underlying knowledge ressources.

A. Application: CCC Challenge

The Computer Cooking Contest is an open competition which started one year ago and was organized during EC-CBR2008 and ICCBR 2009 conferences. It consists in elaborating a software that has to create a recipe for a single dish or even a three course menu.

The overall competition is structured into a main compulsory task and two additional challenging tasks. In this paper we only consider the compulsory task. It involves answering queries that require the selection and – where appropriate – modification of a recipe for a single dish.

The queries to the system are given in natural language and consist of a number of wanted ingredients and other requirements for the dish or menu. These requirements concern the dietary practices, the type of meal or cuisine. A sample query could be to “cook a main dish with turkey, pistachio, and pasta without garlic”. An appropriate answer would be to select a recipe for pistachio chicken and to replace chicken by turkey.

The software evaluation criteria are: (i) culinary quality of the created recipes (appropriate to the query, tasty, cookable, creative); (ii) technical quality of the software (usability, maintainability, performance and scalability; (iii) scientific originality of the approach. According to the CCC rules, the software has to answer to five exercise queries that allow a preliminary evaluation of the different systems. These queries are written in natural language and a main focus can be specified.

Q1: Cook an Asian soup with leek.
(Main focus: type of meal, type of cuisine)
Q2: I would like to have a salad with celery. Please consider that I follow a gout diet.
(Main focus: dietary practice)
Q3: Prepare a low-cholesterol dessert with strawberries and avoid citrus fruits.
(Main focus: dietary practice)
Q4: Cook a risotto with carrots.

1http://www.wi2.uni-trier.de/ccc09/
Q5: I would like to cook a pear pancake.

The software input is a database of basic recipes from which appropriate recipes can be selected, modified, or even combined.

Recipes are textual documents and expressed in a loosely structured XML format (see appendix for an example). Elements are used for identifying the recipe title (TI tag), the ingredients (IN tags), the preparation (PREP tag) is decomposed in several steps (STEP tags).

B. System: TAAABLE

TAAABLE is a Web application involving three main components: the CBR engine, the knowledge base and the interface.

A case-based reasoning (CBR) is a reasoning paradigm that aims at reusing past experiences for solving a new problem [11], called the target problem. A case is often described by a problem and its solution. CBR generally consists in (1) the retrieval of a source case from a case base and (2) the adaptation of this retrieved source case in order to solve the target problem. Textual CBR [5], [6] is a CBR approach in which cases are available in a textual form (or contain a textual part).

A CBR system is a knowledge-based system whose knowledge base is a set of inter-related knowledge containers [10], including the case base, the domain ontology, the case indexes (summarized representation of cases and ), the retrieval knowledge (also known as the similarity and often represented by a parameterized similarity measure) and the adaptation knowledge.

The queries of CCC involve several issues: selecting ingredient, avoiding other, selecting a type of dish (salad, mainDish, etc), precising the dietary practice, etc. Moreover, queries can also involve constraints on the "type of cuisine" (Cook an Asian soup with leek.). The goal of the CBR engine is, given the query, to retrieve and adapt suitable recipes. Adapting a recipe consists in replacing ingredient(s) by other(s). For instance replacing "celery" by "leek". A query is represented by a conjunction of literals based on a vocabulary which is a finite set of propositional variables. In order to adapt a retrieved recipe, one needs adaptation knowledge. The knowledge adaptation is used to perform adaptations and substitutions. The CBR process implemented in TAAABLE basically consist in retrieving a set of suitable recipes and to adapt them if needed [2].

III. MATERIALS AND METHODS

We present below materials that we exploited and the way we conducted this work.

A. Materials

We exploited the expert knowledge of the cooking domain. This knowledge comes from our experience and other knowledge available at websites specialized in cooking, knowledge of alimentary regimes, of dietetic and encyclopedias where classifications are available (see appendix). We also exploited semantic resources used by the TAAABLE system: the ingredient hierarchy and the dish type and origin hierarchies. We describe these hierarchies hereunder.

The ingredient hierarchy (Ingredient.owl) was built manually from the Cook’s Thesaurus and from the CCC database. The Cook’s Thesaurus is a cooking encyclopedia that covers thousand of ingredients, including synonyms and suggested substitutions. At the same time, a terminological base was built in order to associate to each ingredient subconcept a linguistically preferred term as well as a set of morphological variants or synonyms. The ingredient hierarchy and the terminological data base were then manually enriched by adding new concepts and new lexical forms that occur in the recipe data base but do not occur in the thesaurus. Each ingredient component of the recipe data base is linked to an ingredient concept by an annotation process.

![Excerpt of Ingredient.owl](image1.png)

The two hierarchies "Dish Type" and "Dish Origin" have
been built manually and brought together in a resource (DishTypeAndOrigin.owl) made of 160 concepts. Starting from the organisation of dish types and dish origins in the Recipe data base, a list of general dish types and dish origin has been collected, and hierarchically organized. The dish origin hierarchy has two levels. The first level classifies dishes following their mainlands, such as Africa, Asia, Europe, etc. Each first level concept is specialized, on the second level, by the country, origin of the dishes. For instance, Austrian, British, French, German, Italian, etc. are subconcepts of Europe, meaning that, for example, an Italian recipe is also a recipe of Europe. In the same way, the Dish Type hierarchy is mainly organized in two levels. At the first level, there are concepts like BakedGood, Burger, Dessert, MainDish, etc. The second level details, if necessary, the first level concepts. For instance, BakedGood is divided into Bagel, Biscuit, Muffin, Brownie, Cookie, etc. However, these concepts are not detailed more deeply even if more specific categories exist in recipe data base, as it is for example the case for Cookie which is subdivided into Apple Cookie, Chocolate Chip cookies, Diabetic Cookie, etc. These concepts could indeed be defined by the conjunction of being a cookie and by whether or not they contain some specific ingredient.

The annotation process aims at formally representing the content of a recipe as well as defining its categories. This process is in between what is usually called controlled indexing [4] where terms come from a predefined terminology and semantic annotation [12] where terms (named entities, sequences of words) are explicitly associated with the respective and most specific classes in the ontology. The result of the annotation of a recipe is a set of concepts indexing the recipe. Finally the annotation process has to index recipes following the mainland and the country (e.g., Asia, Korean) and following the dish type (e.g., main dish, dessert). All these types are defined in the ontology as concepts.

B. Methods

In order to develop a benchmark that is useful to all systems, we did not begin the evaluation by judging the TAABLE answers. We assessed the whole recipe base for each query. The assessor is supposed to be a user of a system that has knowledge in cooking domain.

a) Building of candidate recipes: The base of recipes contains 1500 recipes, in order to avoid to explore manually each of these recipes, we performed some selection per query of recipes that have to be assessed. The selection of candidate recipes relies on recipe title, ingredients and preparation. We performed a search based on keywords extracted from the query. We enlarge the search by neighbour terms; these terms are set manually by the assessor and are extracted from knowledge material detailed above. The manual establishment of the list of neighbour terms was carefully detailed, assessors had to explicit the knowledge used and the reasoning performed. The aim is to keep traces that are useful for assessors to confront their choices and to update the benchmark in case of knowledge evolution.

The candidate recipes are then recipes that contain at least one of the keywords of the query or its semantic neighbours. The remaining recipes are considered as irrelevant.

b) Relevance judgments: We aim at providing a benchmark inspired by test suites in information retrieval. To measure ad hoc information retrieval effectiveness in the standard way, the test collection consist of three things: (i) a document collection, (ii) a test suite of information needs expressed by queries and (iii) a set of relevance judgments.

The relevance judgments are standardly a binary assessment of either relevant or nonrelevant for each query-document pair. This is not acceptable in our case because a recipe is rarely totally relevant to the query. Furthermore we want to keep trace of each assessor judgment in order to understand the assessor decision and confront it with another decision in case of conflict.

To do so, the relevance of a recipe is estimated by a scale of graduated judgement from 0 to 4. In fact, it is important to distinguish between recipes that totally fit to the query (Totally relevant (TR)) and those that fit partially to the query (partially relevant (PR)). The relevance judgment takes into account the importance of the elements expressed by the query. We distinguish two types of elements: central elements (CE) and peripheral elements (PE) (see table I). Substitutions are performed in priority on peripheral elements.

The comparison between the recipe candidate and the translation of the query in terms of CE and PE leads to these actions:

![Excerpt of DishTypeAndOrigin.owl](image-url)
(i) no modification, exact match, (ii) specialization of one or some elements of the query; (iii) generalization of one or some elements of the query; (iv) modification of one of PE (change to an equivalent element, add or delete a PE). These actions are considered in the estimation of graduated judgment.

<table>
<thead>
<tr>
<th>PE</th>
<th>Relevance</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>0</td>
</tr>
<tr>
<td>N</td>
<td>PR ou TR</td>
</tr>
<tr>
<td>PR</td>
<td>N</td>
</tr>
<tr>
<td>PR</td>
<td>PR ou TR</td>
</tr>
<tr>
<td>TR</td>
<td>N</td>
</tr>
<tr>
<td>TR</td>
<td>PR</td>
</tr>
<tr>
<td>TR</td>
<td>TR</td>
</tr>
</tbody>
</table>

**TABLE I**
GRADUAL SCALE OF RELEVANCE. N: NON RELEVANT, PR: PARTIALLY RELEVANT, TR: TOTALLY RELEVANT

This gradual relevance enables to perform a strict comparison (relevance 4) and relaxed comparison (relevance 3 and 4). The remaining relevance values were exploited by assessors in order to explain their choices and can be modified in case of knowledge evolution or application need changes.

The operations performed to get a relevant recipe are: a substitution of one term of the query, a specialization/generalization or an addition/remove of an ingredient in the recipe.

- replacement of an element of the query by:
  - specialization: a term is replaced by a term that is more specific. For example, *asian soup* is replaced by *chinese soup*. The recipe is considered as totally relevant because it fits the query.
  - generalization: a term is replaced by a term that is more general. The answer can be judged relevant if it is not far from what is expected. This is the case for example for Moroccan dish if replaced by *north African dish* which is more relevant than *African dish*.
  - substitution
    * synonymy or semantic proximity: a term is replaced by an equivalent or substitute one. In case of ingredients, that can be an ingredient with same nutrition value.
    * antonymie: an ingredient is replaced by a term that is its opposite. For example, *high fat milk* can be replaced by *low fat milk*.
- replacement of an element of the recipe by an operation of:
  - adding ingredient: in case of absence of answer even by modification, we can propose to modify the recipe by adding ingredients. For example we can propose to add *carrots* to a recipe of *Herb risotto*.
  - removing ingredient: in case of absence of answer even by modification, we can propose to modify the recipe by deleting ingredients. For example we can propose to remove *citrus fruit* from a recipe in order to avoid them.


**IV. RESULTS**

The set of relevance judgments is used in order to compare system results by using classical measures such as precision and recall. We compare the results of TAAABLE systems with the set of relevance judgments, we consider as relevant the recipes that had a value of relevance upper than 2.

<table>
<thead>
<tr>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>1</td>
</tr>
<tr>
<td>Q2</td>
<td>0.46</td>
</tr>
<tr>
<td>Q3</td>
<td>0.25</td>
</tr>
<tr>
<td>Q4</td>
<td>0</td>
</tr>
<tr>
<td>Q5</td>
<td>1</td>
</tr>
<tr>
<td>Total</td>
<td>0.48</td>
</tr>
</tbody>
</table>

**TABLE II**
PRECISION AND RECALL COMPUTED ON FIRST RESULTS OF TAAABLE

The values of precision and recall can be used to compare the results of TAAABLE with other systems.

In order to deepen the analysis of the results of TAAABLE, we identified the differences between the system results and the set of judgments. Two types of differences are identified and can be explained by:

- the assessor knowledge
  - the assessor has used implicit knowledge not formalized in the system;
  - the assessor has decided a replacement of an ingredient even if it is not an immediate neighbour. This is the case for example with pancakes that can be cooked with pear instead of berries.
  - the assessor has decided a replacement of a preparation by an equivalent one. This is the case for pancake preparation which was replaced by waffle.
- the fuzziness of the resource: this is the case for gout diet which subsumes pulse, while fresh peas and beans have to be authorized.

In some cases, we detected a bug in the system where a constraint in the query is not taken into account. This was the case for example with the query 3 where recipes with citrus fruits were returned by the system.

\[\text{precision} = \frac{|S \cap R|}{|S|} \quad \text{recall} = \frac{|S \cap R|}{|R|}\]

where \(|S|\) is the number of elements retrieved by the system, \(|S \cap R|\) is the number of relevant elements retrieved by the system, and \(|R|\) is the number of elements in the gold standard.
We have done a qualitative analysis of the results in order to explain the silence and noise. This analysis enable the production of the guidelines presented below.

V. GUIDELINES

Some of the system mismatches, in term of noise and silence, can be explained by the accuracy level of the knowledge resources. Therefore, it seems that a substantial improvement of the system performances would be achieved by means of the resource enhancement.

The result analysis leads to focus more precisely on: (i) the gap between terminology and ontology, (ii) the knowledge representation through the ontology (concept and relation definitions and concept structuration). In fact during the adaptation step, the system performs substitution operations for building a new solution. If the resources contain the knowledge represented at a good level of granularity, it's easy to adapt. Therefore, it may happen that the ambiguity, imprecision or lack of the represented knowledge constitute source of mismatches. An other reason is related to the choices made in knowledge representation. For instance in the TAAABLE context, this representation is hardly constrained by the formalism used by the search engine (propositional logic).

A. Gap between terminology and ontology

Selecting a recipe requires to check if the ingredients from the query are explicitly expressed in this recipe. The problem induced by the polysemy of terms may stand in the way of the recipe selection. In this case, the quantity of the given ingredient, the presence of related adjective or the apposition of two ingredients may help to solve the ambiguity. For instance, pepper denotes both the vegetable "pepper" or the "spice". We may sometimes rely on: - a quantity: "1/2 tablespoon of pepper" or "1/2 yellow pepper"; - adjectives: "green pepper" or "freshly ground black pepper"; - a term apposition like "salt and pepper".

In conclusion, a guideline should be to use heavily common-sense heuristics which need an elicitation effort. This may be achieve by adding a set of domain specific rules in order to control term usage.

B. The domain covering by the resources

When we established the benchmark we had used expert knowledge coming from miscellaneous resources such as botanical classification, medical encyclopedia (knowledge about dietary practices), dietetical knowledge (composition tables of the ingredients) or empirical knowledge coming from tricks and tips coming from cooking web site. For example, the equivalence between ingredients like "leek" and "scallion" relies on the classification of the chemical properties of the ingredients. The substitution between "rhubarb and sugar" and "apple" is based on a granny trick. This trick is justified by the need to sweeten the rhubarb acidity. Moreover, the use of the botanical classification may help to decide a substitution. For instance, the knowledge of the pipped fruit class (including apple, blackberry, pear, raspberry, strawberry, etc.) enables to suggest some substitutions between these fruits. Therefore, the system have to rely on some extra knowledge covering these aspects. A guideline should be to ensure that the domain covering by the resources is sufficient. Even if this coverage is insufficient, its measure constitutes a good indicator for selecting the accurate resources.

C. Knowledge representation through the resources

In the TAAABLE system, the available resources are a set of hierarchies. They are composed of non disjoint concepts that may be duplicated. Currently, a diet specification is defined as a high level class in which the ingredients are replicated (see Figure 3).
An evaluation by the intrinsic characteristics of the ontology in regards to the formal constraints must be performed to guarantee the correctness of the reasoning process.

D. Discussion

The evaluation task of knowledge based applications is tightly bound to the quality of the available resources. After the analysis of the previous results, it appears that in addition to the knowledge representation of the ontology, the modularity constitutes an advantage if some resources should be added to the existing ones. Moreover, modularity and reuse appear as factors facilitating the evolution of the resources associated to knowledge based applications. Each of these resources should be evaluated keeping in mind that different techniques of evaluation are dedicated to different goals.

Once this evaluation achieved, an ontology has been built in order to help the conception of the retrieval and adaptation processes of the system. Therefore, the conceptual choice for the ontology development has been strongly driven by the goal of this particular CBR system. The reuse of existing ontologies has been carefully examined but no more considered as these ontologies did not cover what was intended to be reached in this system. Nevertheless some part of the underlining conceptual models were reused to construct the new ontology. The closer ontologies are (i) an ontology of culinary recipes, proposed in [13] and developed to be used in a semantic querying system for the Web; (ii) the Cooking Ontology [9] which grasps four main modules covering the key concepts of the cooking domain: actions, food, recipes, and utensils and three auxiliary modules: units and measures, equivalencies and plate types. The former ontology focused on an ingredient classification, the latter on the top level of a conceptual model of the cooking ontology.

During the building of this ontology, three tasks have been carried out: development of the cooking conceptual model, formalisation of the domain, and implementation in OWL language. Several main classes were identified to elaborate the cooking conceptual model: Recipe, Ingredient, FoodComponent, Food, Action, Amount, and Utensil (see figure 4). These classes were sufficiently independent to decide to build a modular ontology.

This conceptual choice has been made to facilitate the enrichment of the different modules (because the concepts included in these modules are disjoint) without changing the content of the others. The main relations linking these modules have been manually defined. Then the four hierarchies described above were defined.

During the elaboration of the benchmark it appeared that it would be very interesting to take into account the utensil concepts. Indeed, these concepts could help to characterize the type of preparation ("wok" is typical of "asian cooking") or for example to distinguish more easily the difference between dishes ("crock-pot" is typical of "stew" rather than "soup"). Once defined the relation between the modules, the system could rely on them for reasoning and in this way identify the links between the elements useful during the adaptation process.

VI. Related Work

This work focuses on the evaluation of knowledge based application and deals with ontology evaluation problems. Various approaches exist in the literature. We classify them in two categories depending on characteristics of the evaluation: intrinsic and extrinsic [14].

- Many approaches propose to evaluate ontologies by their intrinsic characteristics. (OntoMetric[7], OntoClean[3], O$^2$ and oQual [1], ...)
- Extrinsic evaluation depends on which artifact to compare with and on the application context. We can classify them into two categories: comparison to a reference and application-based comparison.

This work can be classified in the latter category. Part of the success of IR as a field came from the use of established, well-maintained, and almost universally accepted benchmarks for testing the work of IR methods. Benchmarks for ontology evaluation exist but they are interested to test scalability, performance tradeoffs and persistance [8]. To our knowledge there is no IR-like benchmark for applications that use semantics.

VII. Conclusion

We propose a methodology to evaluate knowledge based applications. We setted up a benchmark that is useful to the system we evaluated and can be used by other systems in CCC challenge context. The comparison of the system results with the benchmark enables to perform a quantitative and qualitative analysis of the semantic resources used and the system reasoning.

The values of precision and recall can be used to compare the results of TAAABLE with other systems. Unfortunately, the CCC expert judgments are not stored. The assessments

![Fig. 4. Excerpt of the cooking conceptual model](image-url)
are user appreciations and we can not say why a system results were preferred to another. We aim at applying our propositions in the context of the CCC next year, it would also be interesting to compare expert assessments with ours.

We hope that the benchmark contributes to have a common base test of evaluation driven by application. This benchmark can be used by any informal retrieval system (not necessarily applying CBR) that takes knowledge into consideration. It is helpful to detect noise and silence and to isolate problems which are related to the knowledge base.

ACKNOWLEDGMENT

The authors would like to thank Rim Bentebibel (LIPN), Catherine Duclos (LIM&BIO) and Sylvie Salotti (LIPN) for fruitful discussions. This work was partly realized as part of the Quaero Programme, funded by OSEO, French State agency for innovation.

REFERENCES


