Conversational Retrieval of Cooking Recipes

Christian Zeyen, Gilbert Müller, and Ralph Bergmann

Business Information Systems II
University of Trier
54286 Trier, Germany
[zeyen][muellerg][bergmann]@uni-trier.de,
http://www.wi2.uni-trier.de

Abstract. This paper presents a new approach for exploring a collection of cooking recipes represented as cooking workflows by means of a conversation. Users are guided through the search process by answering posed questions. Thus, they are not required to formulate queries and they do not need to browse a recipe collection by hand. Questions involve ingredients and cooking activities contained in the workflows. The approach is implemented in our CookingCAKE system, extending it with a dialog component.

Keywords: Recipe Retrieval, Conversational Retrieval, Workflows

1 Introduction

Nowadays, numerous cooking recipes are available online and various search engines support users in finding suitable recipes. Beside providing a keyword-based search, some engines also support an ingredient-based search by asking the user to specify desired and undesired ingredients. However, in practice, amateur chefs may only have a vague idea of their desired dish or they lack detailed knowledge about required ingredients and thus have difficulties in providing a precise query. In 2010, Yummly[1] launched the first semantic search platform for food and recipes. By capturing the semantics of recipe descriptions and ingredients, Yummly is able to handle more vague queries such as general terms in a keyword-based search. For example, if the user starts a search with the keyword *meat*, Yummly initiates a dialog asking the user which kind of meat she would like. Then, further questions are posed concerning various properties such as the desired type of dish, preparation time, nutritional preferences, and additional ingredients.

This paper follows a similar approach and provides a method to conduct a conversation with the user to find desired cooking recipes. We focus on structural features of recipes thus representing them as workflows. More precisely, in addition to considering the occurance of ingredients and preparation steps, we also analyze their dependencies. Based on this information, questions concerning the further processing of desired ingredients can be posed. Thus, the ingredient-based search capabilities provided in typical search engines for recipes are extended by this approach. Moreover, we investigate how such features can be constructed automatically from the underlying workflow

[1] www.yummly.com

repository and we propose a respective question selection strategy for the conversation. We consider a recipe search as a problem solving process in which the problem description specifies the user's preferences of a desired dish and a possible solution is a recipe describing its preparation. We apply the methodology of conversational case-based reasoning (CCBR) [1,2], which particularly focuses on the interactive nature of problem solving. CCBR approaches include methods which incrementally elicit the relevant features of the target problem in an interactive dialog, often with the aim of minimizing the communication effort for the user. The basic assumption behind CCBR is that guided question answering requires less domain expertise than providing detailed queries from scratch. To apply CCBR with cooking workflows, we combine CCBR with process-oriented case-based reasoning (POCBR) [5], which usually deals with cases as workflows or process descriptions expressing procedural experiential knowledge. Consequently, we propose a new conversational POCBR approach [10], called C-POCBR, for the retrieval of cooking workflows. We implemented the approach in our CookingCAKE system [6], which is part of the CAKE framework[2], extending it with a dialog component. CookingCAKE is a POCBR system for retrieving and adapting[3] cooking workflows based on a user-defined query specifying desired and undesired ingredients and preparation steps.

In the following, section 2 briefly introduces the representation and querying of cooking workflows before section 3 describes our C-POCBR approach. Section 4 concludes the paper and briefly discusses future work.

2 Cooking Workflows

In our approach a cooking recipe is represented as a workflow describing the process to prepare a particular dish [9,3] (see Fig. 1). Cooking workflows consist of a set of *preparation steps* (also called *task nodes*) and a set of *ingredients* (also called *data nodes*) shared between its tasks. Task nodes are linked by *control-flow edges* defining the execution order. This forms the *control-flow*. Task nodes, data nodes, and relationships (represented by *data-flow edges*) between the two of them form the *data-flow*. To each

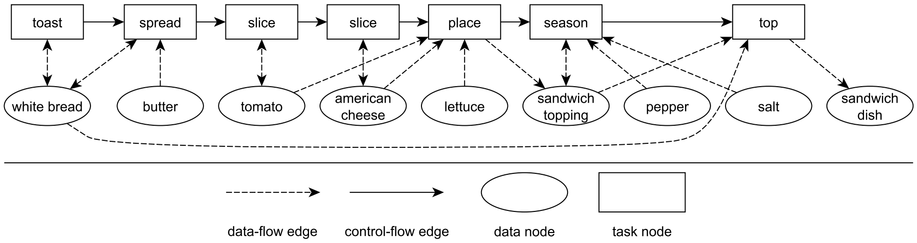


Fig. 1. Example of a Cooking Workflow

[2] See cake.wi2.uni-trier.de

[3] However, the dialog component does not yet consider the available adaptation methods.

node a semantic label is assigned, which is structured hierarchically in a data taxonomy of ingredients or a task taxonomy of cooking steps. Thereby, workflows can be generalized regarding their semantic labels [7]. Generalized workflows provide a more general description and thus stand for a set of more specific workflows. For example, the workflow in Figure 1 can be generalized by generalizing the ingredient *american cheese* to the more general ingredient *cheese* from the data taxonomy.

In order to retrieve cooking workflows with case-based reasoning, the user's preferences must be specified in a query, which in turn must be evaluated against the available workflows. For this purpose, we proposed a process-oriented query language (POQL) [8] and a similarity measure [10] to determine the best-matching workflows for a given POQL query. In a nutshell, POQL consists of two parts. In a *query part*, the user can specify workflow fragments representing properties the searched recipe should fulfill. In an additional *restriction part*, the user can define undesired situations, e.g., unwanted ingredients, which should be avoided.

3 A Conversational Retrieval Approach

To facilitate the elaboration of a POQL query for workflow retrieval, our approach guides the user through the query process with a sequence of questions about her preferences. The more questions are answered, the more knowledge about desired and undesired properties is available, which is stored in an internal POQL query. A major focus is put on the automatic creation of questions to avoid that they need to be specified manually. For this purpose, we consider workflow fragments as characteristic properties of a workflow, which we refer to as features. The basic idea is to extract features from the workflows stored in the repository (case base) automatically, which are then used as the subject of questions. In order to conduct efficient conversations, we rank features by their ability to distinguish workflows from one another. Furthermore, identified relations between features enable to generate coherent follow-up questions and to infer irrelevant features based on already answered questions.

3.1 Features of a Cooking Workflow

In principle, a feature can be any fragment of a cooking workflow. In a workflow, the smallest possible feature consists of a single workflow item. This can be a single node such as a data or a task node. More complex features can be created by extracting partial workflows. To derive questions on a more general level of detail, we apply a generalization algorithm [7], which generalizes semantic labels based on the task and data taxonomies. The generalization produces a generalized workflow W^* from the original workflow W, from which more general features can be extracted. We extract and annotate two different kinds of features for each workflow W in the case base:

- *specific feature nodes* and *generalized feature nodes*, i.e., single nodes from W and single nodes for all generalizations within the taxonomy up to the respective node in the generalized workflow W^*
- *specific feature workflows* and *generalized feature workflows*, i.e., partial workflows (consisting of more than one node) from W and W^*, respectively

A *feature workflow* W_d describes structural properties of a workflow W regarding a particular data item d from W. It consists of all (and at least one) tasks connected to d and which are connected by control-flow edges. Moreover, W_d additionally comprises all data items that are connected to those tasks. For instance, regarding the cooking workflow depicted in Figure 1, a feature workflow constructed for the ingredient *white bread* contains the associated processing steps *toast* and *spread*. It further comprises the ingredient *butter*, which is required for the task *spread*.

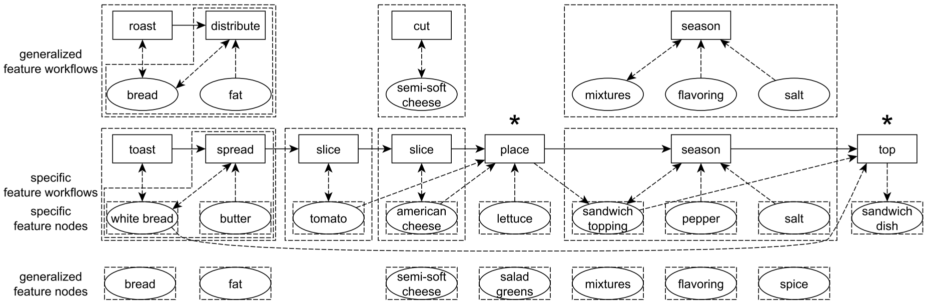


Fig. 2. Examples of a Workflow's Features

Figure 2 exemplifies all features (see dotted rectangles) extracted from the cooking workflow depicted in Figure 1 . The specific workflow is depicted in the middle of the figure. Related features (such as specific and generalized features) are arranged near one another. For instance, the specific feature node *pepper* is related to the generalized feature node *flavoring*. Based on the taxonomy, an additional generalized feature node *spice* laying inbetween those two is extracted as well.

With respect to the cooking domain, we applied some domain-specific restrictions. For the feature extraction, we omit single task nodes as they are mostly of no relevance when considered on their own. In addition, to obtain easy-to-understand feature workflows, we exclude tasks (marked with "$*$") that produce new data by consuming other data.

In a second step all extracted features are sorted in descending order by their ability to distinguish workflows from one another. By this means, we reduce the length of a conversation. We adopt the simVar measure by Kohlmaier et al. [4], which utilizes the similarity variance as a ranking criterion. It estimates the variance of the similarity of the most similar workflows assuming that the value of the respective feature in the query is known (see [10] for more details).

In the next step, relations between features are analyzed. For each feature f all related features are determined. The set of related features of a feature f contains those features g that share a common partial workflow with f which is either a generalization of f or g. Related features can be differentiated by their number of nodes and by their generality of nodes. A feature may have related features that are *larger*, *equally large*, or *smaller* as well as related features which are *more specific*, *equally specific*, or *more general*. For example, for the feature workflow $f_1 = \{$*slice, ham*$\}$, the related feature $g_1 = \{$*cut, meat*$\}$ is *more general* and *equally large* while the feature $g_2 = \{$*parma-ham*$\}$ is *more specific* and *smaller*.

3.2 Questioning Strategy

In order to obtain the user's preferences most suitable to determine the best matching cooking workflows, i.e., the candidate workflows, a respective questioning strategy is required. The dialog component iteratively creates and displays questions until the user selects a desired workflow. With each question answered by the user, the set of candidate workflows, which encompasses the whole case base at the beginning of a conversation, is reduced. The dialog starts with an empty query and the set of candidate features, i.e., relevant features to be asked in a question, comprises the full set of features.

In the main loop, the dialog component selects a question based on the candidate features. The selection process considers the previously answered questions as well as the ranking and relationships of features. Each question involves one or in certain cases several candidate features. We provide three major types of questions, which are depicted in Table 1. Based on the ranking of the candidate features, the subject matter of a question is determined. If the user answers that the suggested feature is desired, specific follow-up questions are selected in the subsequent iterations. Those follow-up questions are derived from related features and aim at further refining the previous question asked. An example is given in Table 1, which presents a sequence of three questions (Q) including the possible answers (A) and the user-selected answers (marked with a box).

Table 1. Question Sequence in a Conversation

Order	Question Type	Subject Matter	Example
1.	initial feature question (FQ)	highest ranked feature	Q: Is $\{meat\}$ a desired feature? A: $\boxed{\text{desired}}$, undesired, irrelevant
2.	follow-up specialization question (SQ)	more specific feature(s)	Q: Is there a suitable specialization for $\{meat\}$? $\{poultry\}, \{ham\}, \boxed{\{\text{chicken}\}}, \ldots$ A: $\boxed{\text{apply}}$, select undesired feature(s), irrelevant
3.	follow-up enlargement question (EQ)	larger feature(s)	Q: Is there a suitable enlargement for $\{chicken\}$? $\boxed{\{\text{shred, chicken}\}}, \boxed{\{\text{chop, chicken}\}}, \ldots$ A: apply, $\boxed{\text{select undesired feature(s)}}$, irrelevant

At the beginning of a conversation the highest ranked feature from the candidate features is suggested in a feature question (FQ). This type of question is not related to previously suggested features and it will be asked as long as the user selects the suggested feature as irrelevant or undesired. In the example, the feature *meat* is the subject matter of the first question and it is selected as *desired* by the user.

If a feature in a FQ is selected as desired, a first follow-up question, i.e., a specialization question (SQ), is posed suggesting one or (if available) several equally large but more specific features. The user can choose a specialization, select specializations as undesired, or mark all specializations as irrelevant. This type of question is repeated as long as further specializations exist and are desired by the user. In the example, the user chooses *chicken* as her desired type of *meat*.

Following the SQs, an enlargement question (EQ) is displayed to the user that suggests, if available, larger features than the previously selected and/or specialized feature. Just as in SQ, the user has three different options: choose an enlargement, select enlargements as undesired, or mark all enlargements as irrelevant. In the given example, we assume that the user does not like *shredded* or *chopped chicken* and thus selects those features as undesired. If no more EQs are available, the next initial FQ is selected, addressing a new and potentially unrelated subject matter.

When the set of candidate features is updated due to an ignored or answered question, irrelevant features are inferred based on the relations between features. If a question is marked as irrelevant, all the related features (e.g., more specific and larger features) are marked as irrelevant, too. If suggested features are selected as undesired, they are added to the restriction part of the current query and related irrelevant features are no longer considered as candidate features, to prevent the system from repetitively asking the user what she does not like. If a feature is marked as desired, also related features such as more general features are removed from the set of candidate features. If a user chooses a specialization or an enlargement, the target feature that is already present in the query is replaced with the new feature.

3.3 Conversation with CookingCAKE

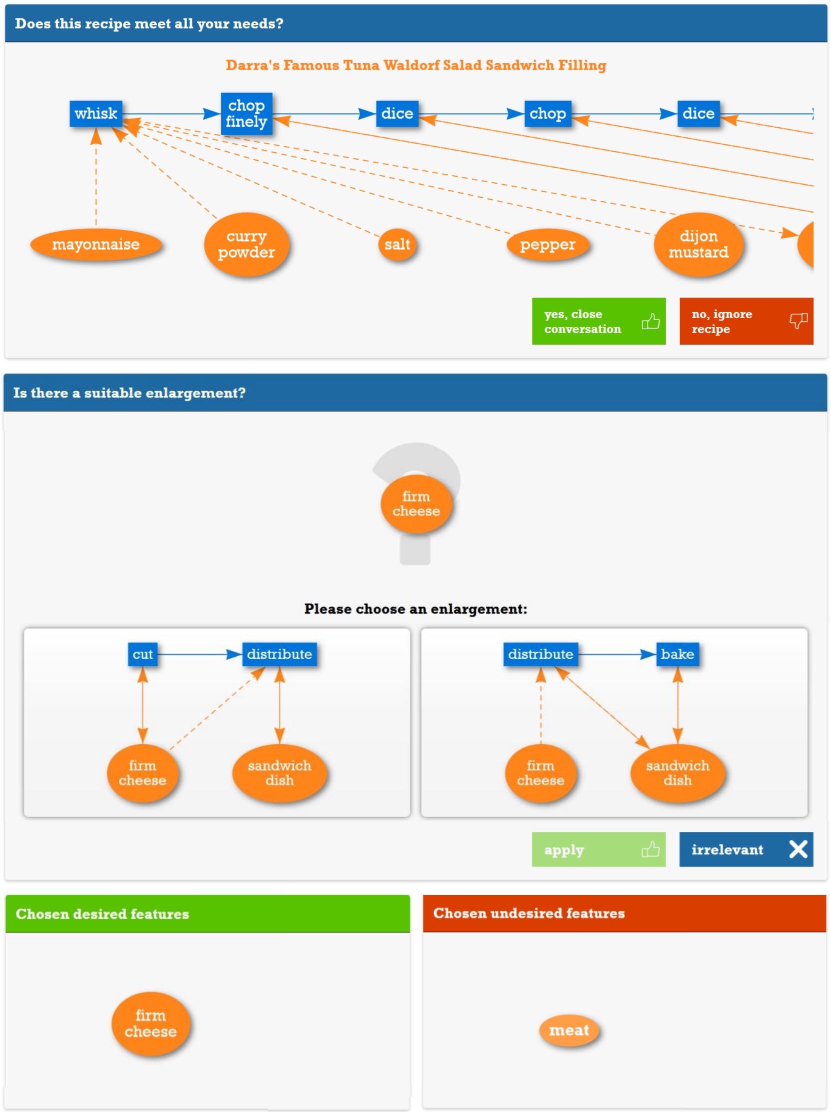
Based on the extracted features and the questioning strategy, the conversation is conducted in the dialog component of CookingCAKE[4]. The graphical user interface is illustrated in Figure 3. It consists of three displays suggesting the best matching workflow (upper part of figure), showing a question (middle of figure), and summarizing the current query (lower part of figure). Figure 3 presents a progressed state in a conversation in which some preferences are already obtained from the user and specified in the internal query by the system. In the given example, the current query contains *firm cheese* as desired and *meat* as undesired. The question displayed is a follow-up question targeting the further refinement of the current query. In the example, the question suggests alternative processing steps for *firm cheese*. The user has two options to react:

1. *Ignore the question:* In this case, the features being subject of the question as well as related features are ignored and the next best question is displayed.
2. *Answer the question:* Causes the system to extend the query and to perform a similarity-based retrieval on the current set of candidate workflows. The workflow with the highest similarity is displayed to the user.

In the upper part of the user interface, a solution workflow best fulfilling the current query is suggested to the user. With respect to this suggestion, the user has two additional options to react:

[4] Online demo available at `cookingcake.wi2.uni-trier.de/conversation`

1. *Ignore the suggestion:* The user can actively ignore the suggested workflow, which causes the system to exclude it from the solution candidates and to trigger a new retrieval for the next best workflow.
2. *Select the suggestion:* In this event, the conversation terminates successfully.



Does this recipe meet all your needs?

Darra's Famous Tuna Waldorf Salad Sandwich Filling

Is there a suitable enlargement?

Please choose an enlargement:

Chosen desired features

Chosen undesired features

Fig. 3. Dialog Component of CookingCake

4 Conclusions and Future Work

We presented an approach to retrieve cooking workflows by means of an interactive dialog with users. To save effort for defining suitable questions, a method for the automatic creation of questions based on extracted features was described. We recently showed in an experimental evaluation that those features are meaningful subjects of questions and that they are suitable to distinguish workflows from one another [10]. Furthermore, our results indicate that the conversational approach has the potential to reduce the retrieval time and thus is able to reduce the communication effort for users.

In future work we plan to extend the presentation and explanation of workflows and features. For the sake of simplicity, we used a simplistic representation of cooking recipes and considered basic features, which could be extended in the future. Also, future work should investigate how adaptability of workflows can be considered during a conversation. By this means, interactive retrieval could be combined with interactive adaptation to provide more diverse and customized cooking workflows for users.

Acknowledgments. This work was funded by the German Research Foundation (DFG), project number BE 1373/3-3.

References

1. Aha, D.W., Breslow, L.A., Muñoz-Avila, H.: Conversational case-based reasoning. Applied Intelligence 14(1), 9–32 (2001)
2. Aha, D.W., McSherry, D., Yang, Q.: Advances in conversational case-based reasoning. Knowledge Engingeering Review 20(3), 247–254 (2005)
3. Bergmann, R., Gil, Y.: Similarity assessment and efficient retrieval of semantic workflows. Inf. Syst. 40, 115–127 (2014)
4. Kohlmaier, A., Schmitt, S., Bergmann, R.: A similarity-based approach to attribute selection in user-adaptive sales dialogs. In: Aha, D.W., Watson, I. (eds.) Case-Based Reasoning Research and Development, ICCBR 2001. LNCS, vol. 2080, pp. 306–320. Springer (2001)
5. Minor, M., Montani, S., Recio-Garca, J.A.: Process-oriented case-based reasoning. Information Systems 40, 103 – 105 (2014)
6. Müller, G., Bergmann, R.: CookingCAKE: A framework for the adaptation of cooking recipes represented as workflows. In: Kendall-Morwick, J. (ed.) Workshop Proceedings from (ICCBR 2015). CEUR, vol. 1520, pp. 221–232. CEUR-WS.org (2015)
7. Müller, G., Bergmann, R.: Generalization of workflows in process-oriented case-based reasoning. In: Russell, I., Eberle, W. (eds.) Proceedings of the 28th Int. Florida Artificial Intelligence Research Society Conference, FLAIRS 2015. pp. 391–396. AAAI Press (2015)
8. Müller, G., Bergmann, R.: POQL: A new query language for process-oriented case-based reasoning. In: Bergmann, R., Görg, S., Müller, G. (eds.) Proceedings of the LWA 2015. CEUR Workshop Proceedings, vol. 1458, pp. 247–255. CEUR-WS.org (2015)
9. Schumacher, P., Minor, M., Walter, K., Bergmann, R.: Extraction of procedural knowledge from the web. In: Workshop Proceedings: WWW'12. Lyon, France (2012)
10. Zeyen, C., Müller, G., Bergmann, R.: Conversational process-oriented case-based reasoning. In: Proceedings of ICCBR 2017. LNCS, Springer (2017)