Opportunistic Adaptation Knowledge Discovery

Fadi Badra$^1$  Amélie Cordier$^2$  Jean Lieber$^1$

$^1$LORIA (CNRS, INRIA, Université de Nancy)
BP 239, 54506 Vandœuvre-lès-Nancy, France
{badra,lieber}@loria.fr

$^2$LIRIS CNRS UMR 5202, Université Lyon 1, INSA Lyon, Université Lyon 2, ECL
43, bd du 11 novembre 1918, Villeurbanne, France
Amélie.Cordier@liris.cnrs.fr
Introduction

Knowledge discovery (KD) techniques have proved to be useful to learn adaptation knowledge, but several issues remain to be tackled.

- When should the KD process be triggered?
  
  Acquiring adaptation knowledge offline forces the system designer to anticipate the need for adaptation knowledge, which can be very tedious.

- On what training set?
  
  Only very general adaptation rules can be learned from an unrestricted training set. More fine-grained adaptation rules are “hidden” in a large set of results. Not restricting the training set may also lead to computational difficulties.

- With which hypothesis space?
  
  Leaving the hypothesis space unrestricted leads to the generation of too many results to enable validation by the expert.
Our proposal: generating adaptation knowledge “on the fly” from the case base

Adaptation knowledge is learned:

1. from the case base by the means of knowledge discovery techniques,
2. online and in an opportunistic manner during a particular problem-solving session.
WikiTaaable is a CBR system in the cooking domain.

Example: “I want to cook a pear pancake”

srce = pancake ∧ pome_fruit  
Sol(srce) = R  
“Apple pancakes from the townships”

pome_fruit ≅≈ pear  
σ = apple ⇝ pear

 tgt = pancake ∧ pear  
Sol(tgt) = σ(R)  
Replace apple by pear in the recipe “Apple pancakes from the townships”
CABAMAKA is a knowledge discovery process that learns adaptation knowledge from the comparison of two sets of source cases (recipes)

\[ R_k = \text{dessert} \land \text{pancake} \land \text{apple} \land \text{pome fruit} \land \text{fruit} \land \text{cinnamon} \land \ldots \]
\[ R_\ell = \text{dessert} \land \text{pie} \land \text{pear} \land \text{pome fruit} \land \text{fruit} \land \text{vanilla} \land \ldots \]

representation of variations

\[ \Delta_{k\ell} = \{\text{dessert}^\equiv, \text{pancake}^-, \text{pie}^+, \text{fruit}^\equiv, \text{pome fruit}^\equiv, \text{apple}^-, \text{pear}^+, \text{cinnamon}^-, \text{vanilla}^+\} \]

generalization

\[ \Delta = \{\text{dessert}^\equiv, \text{apple}^-, \text{cinnamon}^-, \text{pear}^+, \text{vanilla}^+\} \]

interpretation

\[ \sigma = \text{dessert} \land \text{apple} \land \text{cinnamon} \rightsquigarrow \text{dessert} \land \text{pear} \land \text{vanilla} \]

"In dessert recipes, apples and cinnamon can be replaced by pear and vanilla."
Overview of the knowledge acquisition process

Adapt

Test solution

satisfied?

yes
Retain learned adaptation knowledge

no

Explain adaptation failure

Choose repair strategy

Discover new knowledge

Learned adaptation knowledge is made available to the system.
Adaptation failure explanation

pear is incompatible with cinnamon in the obtained recipe!

Failure explanation: a 3 step process

1. select an adaptation operation $\sigma = A \rightsquigarrow B$ to repair
2. choose an explanation pattern
3. instantiate this pattern

- $\sigma = \text{apple} \rightsquigarrow \text{pear}$, $A = \text{apple}$, $B = \text{pear}$
- pattern = “an ingredient $x$ of $B$ is incompatible with an ingredient $y$ of the obtained recipe”
- $x = \text{pear}$, $y = \text{cinnamon}$
Choose a repair strategy

explanation pattern

“*pear is incompatible with cinnamon in the obtained recipe*”

Choose repair strategy:

- [ ] remove cinnamon
- [x] find a substitute for cinnamon (using CabamakA)

repair strategy

\[ \sigma' = \text{apple} \land \text{cinnamon } \rightsquigarrow \text{pear} \land <\text{something?>} \]
How is the KD process parameterized?

\[ \sigma' = \text{apple} \land \text{cinnamon} \leadsto \text{pear} \land \text{<something?>} \]

Filtering variation \( \Delta_{TS} \)

\[ \Delta_{TS} = \{\text{apple}^-, \text{cinnamon}^-, \text{pear}^+\} \]

Filtering the training set

The training set is made of variations \( \Delta_{k\ell} \) between recipes such that \( \Delta_{TS} \subseteq \Delta_{k\ell} \)

Filtering the results

To keep the generalizations \( \Delta \) such that:

1. \( \Delta \) satisfies the constraints expressed by the repair strategy (\( \Delta_{TS} \subseteq \Delta \))
2. \( \Delta \) corresponds to a substitution that is applicable to modify \( \text{Sol(srce)} \)
Repaired adaptation

\[ \sigma' = \text{apple} \land \text{cinnamon} \rightsquigarrow \text{pear} \land \text{vanilla} \]

srce = pancake \land \text{pome\_fruit} \quad \gamma = \text{pome\_fruit} \leftrightarrow \text{pear} \quad \text{tgt} = \text{pancake} \land \text{pear}

Sol(srce) = R \quad Sol(tgt) = \sigma(R)

"Apple pancakes from the townships"

Replace apple by pear and cinnamon by vanilla in the recipe "Apple pancakes from the townships"
Conclusion and future work

- **semi-automatic** approach for adaptation knowledge acquisition
- a KD process is triggered **online** as part of an **interactive** and **opportunistic** knowledge acquisition process
- tested on examples and implemented as part of the WIKITAAABLE system (this year’s *Computer Cooking Contest* challenger!)
- future work include running the KD process on different (and larger) knowledge sources, not only the case base\(^1\).
- some more extensive testing is required to evaluate the approach.

---

\(^1\)The implementation proposed for the CCC already uses a different knowledge source: the KD process learns adaptation knowledge out of a set of 87,000 recipes extracted from the website http://www.recipesource.com.