Conversational CBR

CBR systems that engage in a dialogue (conversation) with users

Used for shallow diagnosis, product selection, planning.

Main money earner for CBR

*CBR-lite*
  - just similarity based retrieval
  - no revision
Conversational CBR

- Pioneer was Inference Corp. with CBR-Express (1993) (copied by others)
  - System lets user create a natural language query
  - Query is matched against case descriptions
  - Matching cases retrieved and confirming questions asked of user (conversation)
  - Uses information gain to order questions
  - Conversation stops when a case’s confidence passes a threshold value
- Case are “authored” not “programmed”

Conversational CBR

- Demo of Help!CPR from The Haley Enterprise
  - www.haley.com

Conversational CBR

- Cases comprise
  - Textual description
  - Set of confirming questions
  - Solution or Action
- Natural language query is matched against case descriptions
  - Uses simple textual comparison
  - Keywords, trigram matching
  - Lexicons, thesauruses & concept hierarchies
Trigram matching

- Word are broken into sets of three letters
  - paper = [[p,a,p],[a,p,e],[p,e,r]]
  - piper = [[p,i,p],[a,p,e],[p,e,r]]
    - approx. similarity = 0.88
- Very good for simple typos
- Letter transposition

Trigram matching

- A trigram space is constructed from all possible trigram combinations of uppercase letter (26^3)
- A string corresponds to a point in the trigram space where its position along each of the 26^3 dimensions represents a constituent trigram
- A sophisticated indexing technique called “Probably Close” indexing folds this high dimensionality down into an index of lower dimensionality (cf. trigram matching.pdf)

Conversational CBR

- Similar cases are retrieved based upon similar descriptions
- An information gain algorithm calculates which question best discriminates between the cases
- A question’s answer alters the confidence (score) of a case
- Questioning continues until a case’s confidence exceeds a threshold
- Or all cases have been excluded
Conversational CBR

- Cases are authored not programmed
- Unlike simple rule-based systems diagnostic systems
- Anyone who can use MS Excel can author cases.
- Significant advantage to end-user companies

Conversational CBR

- Advantages over decision trees
  - Questions are not presented in a fixed order
  - Order is dynamically created at query time
  - Questions do not have to be answered
  - Solutions are available at all times (any-time problem solver)
  - Editing & creating cases is simple – they are independent of each other (unlike a decision tree)

Case representation

- Representation depends on:
  - Requirements of domain and task
  - Structure of available case data
- Flat feature-value list (like a database record)
  - Simple case structure is sometimes sufficient for problem solving
  - Easy to store and retrieve in a CBR system
  - Suitable for shallow technical diagnosis, product recommendation
Case representation

- Object-oriented representations
  - Case: collection of objects (instances of classes) in the sense of OO
  - Required for complex and structured objects
- For special tasks:
  - Graph representations: case = set of nodes and arcs
  - Plans: case = (partially) ordered set of actions
  - Predicate logic: case = set of atomic formulas

Flat case representations

Cases are typically stored in a database

Two types of case-base

- Homogenous
- Heterogenous

Homogenous case-bases:

- All cases contain the same feature-value pairs (note some values may not be known)
- ie they share the same record structure
  - Houses in a real estate case-base
  - Cars in a car dealer’s case-base
- Easy to define a full and sufficient set of case features
Case representation

- Heterogenous case-bases:
  - Cases share some feature-value pairs
  - Cases have some different feature-value pairs
  - Difficult/impossible to define a full and sufficient set of case features
    - Medical record: impossible to define all possible symptoms, tests, diseases, treatments, etc...a person may have

Case representation

- OO representation (structural CBR)
  - A case consists of a set of objects
  - Objects described by a set of features
  - Relations between objects (e.g. part-of)
  - Each object belongs to an object-class.
  - Object-classes are organized in an inheritance hierarchy.
  - Case representation language CASUEL (developed by INRECA project)
Case vocabulary

- Case vocabulary means the features that describe a case
- Sometimes called the case index
- Do NOT confuse with indexing cases for computational efficiency
- Case features (vocab.) should be:
  - Predictive (ie relevant to retrieval)

Case features are:
- Indexed
- Unindexed

Indexed features are:
- used for retrieval
- are predictive of the case’s solution

Unindexed features are:
- not used for retrieval
- not predictive of the case’s solution
- provide valuable contextual information and lessons learned
- Features can change status
Similarity ???

- Purpose of similarity, either:
  - Select cases that can be adapted easily to solve the current problem
  - Select cases that have (nearly) the same solution to the current problem
- Basic assumption:
  - similar problems have similar solutions

Similarity ???

- Degree of similarity = utility or reusability of the solution
- Similarity is an a-priori approximation of reusability
- Goal of similarity modelling:
  - provide a good approximation
  - close to real reusability
  - And easy to compute

Similarity ???

- Assumptions
  - 2 similar problem descriptions have similar solution descriptions
  - It is easier to adapt the solution of a similar problem than the solution of a less similar problem
**Similarity ???**

- **Assumptions**
  - 2 similar problem descriptions have similar solution descriptions
  - It is easier to adapt the solution of a similar problem than the solution of a less similar problem

---

**Problem Space**

- Input problem description

**Solution Space**

- Description of new problem to solve
- Description of solved problems
- Stored solutions
- New solution created by adaptation

---

**Modeling similarity**

- Different approaches depending on case representation
- Similarity measures (metrics):
  - Functions to compare two cases: \( sim(Case_1, Case_2) \) \([0..1]\)
  - Local similarity measure: similarity on feature level
  - Global similarity measure: similarity on case or object level
  - Combines local similarity measures
  - Takes care of different importance of attributes (weights)
Modeling similarity

- Different approaches depending on case representation:
  - (Sub-)Graph isomorphism for graph case-representations
  - Logical inferences