

Case-Based Reasoning

Characteristics

- **Retrieve and Adapt** rather than Decompose and Recompose
- Used in both **problem solving** and **interpretation**
- **Learning** is crucial (especially learning from failure)

Learning is a byproduct of reasoning

Learn new cases

Learn indexes

Generalized Cases

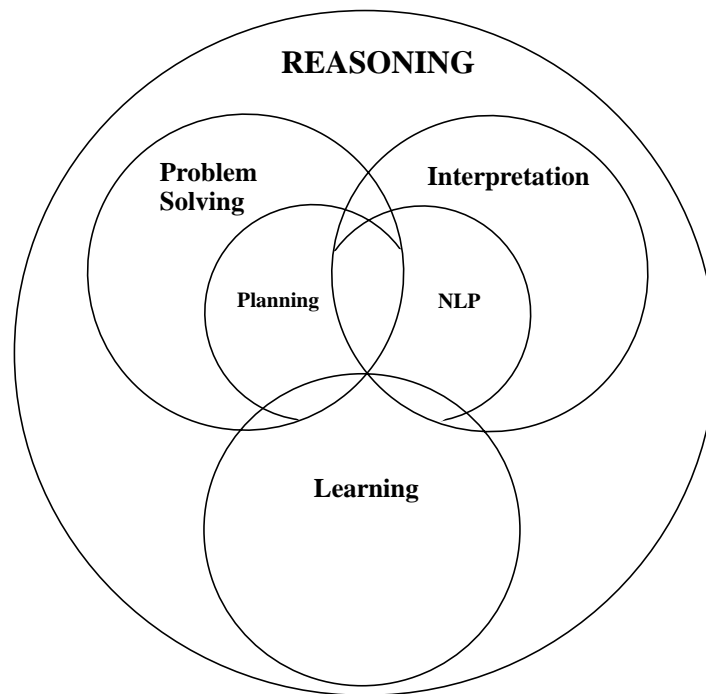
- Successful in

Applications (e.g., product help desk systems)

Cognitive Modeling (e.g., Script theory)

CBR is Different than Traditional AI Approaches

1. Emphasis on memory organization
2. Emphasis on failure-driven learning
3. Emphasis on concrete episodes
4. Emphasis on human behavior
5. Emphasis on weak domains



Problem Solving, interpretation, and learning
are all central to case-based reasoning

- Reasoning is **NOT** just computing a solution
- Interpretation is Understanding

But what is understanding?

What does it mean to understand something?

A movie? A book? The behaviors of others?

Algorithm Pseudocode

- Remember old solution
- Adapt to new situation
- Critique result
- Evaluate execution
- Store solution

What is a case?

- *Contextualized* piece of knowledge representing an *experience* that teaches a *lesson* fundamental to achieving the *goals* of the reasoner.
- The Lesson
 - Associates problem with solution
 - Associates outcome with situation
- The Context (Index)
 - Goal indexing
 - Failure indexing (anticipation)
 - Surface features versus semantic features

The Indexing Problem

- How to retrieve (store) based on what knowledge is needed in the current (future) situation.
- Index is *functional* not surface feature

Functional means according to the *Purpose* of the case

Case Retrieval

- Partial Matching and Similarity Judgements
- Retrieving Complex Recursive Structures Vs. Bit Vectors
- Retrieve *Set* of Similar Cases
- Choose the Best Match in the Set

Adaptation

- “Fixing up” old solution in light of a new situation
- How?
 1. Insertion
 2. Deletion
 3. Substitution

Storage Issue - What cases to store?

- Predicting usefulness of a case at storage time or reinforce useful memories at execution time?
- **Store** if
 - not easily inferred *and*
 - Useful** (teach a lesson) *or*
 - Different** from expectation
- **Forgetting** - *use it or lose it* principle.