Rational and Affective Linking Across Conceptual Cases - without Rules

Graham A. Mann

Artificial Intelligence Laboratory
School of Computer Science & Engineering
University of New South Wales
Sydney, NSW 2052, Australia.
mann@cse.unsw.edu.au

Abstract. Human reasoning across experiential cases in episodic memory seems quite different from conventional artificial reasoning with conceptual representations by systematically manipulating them according to logical rules. One difference is that in humans linkages between particular experiences can apparently be made in a number of qualitatively different ways, forming recollective chains along different dimensions. For example, watching one movie may recall another which had a similar ending, cinematography, or common actors. It may also recall an otherwise unrelated movie which produced the same emotional impact. These linkages do not appear to be economically or simply described by rules. Yet case-based reasoning systems could benefit from sequential indexing of this kind. A conceptual-graph-based FGP (Fetch, Generalise, Project) machine using a small database of intellectual property law cases could enable such “memory-walks” to be computed without rules.

This research was supported by Australian Research Council grant A49600961 "Providing Sophisticated Access to Legal Information" (Gedeon, Greenleaf & Mowbray, 1996-1998).

1 Introduction

Inference engines that manipulate expressions by means of deductive logic can accomplish useful work inside expert systems and knowledge based systems, but operations of this sort seem qualitatively different from a certain kind thinking which seem valuable to humans. A good part of our problem-solving evidently draws on our store of experiences in episodic memory, through which our attention makes excursions that are sometimes called “memory-walks”. Whether in well-organised, goal-oriented, rational trains of thought or in free associations with one idea recalling another apparently at random, these memory-walks are a familiar part of mental life.

The mechanisms by which one experiential pattern may summon another, somehow similar pattern has also been of interest since the origins of case-based reasoning (CBR) in the late 70s. “Reminding” has been a recurring topic in memory-oriented views of conceptual processing [10]. More recently, Gelernter has articulated a theory of “musing” to account for the memory walk phenomenon [9]. The theory conjectures that human minds link ideas together in different ways according to a continuum of
mental focus, ranging from strictly constrained and quite logical operations on specifically selected ideas at the high end, through looser, more idiosyncratic and general connections between groups of related ideas in the middle range, to affect-linked or random "daydreaming" across the entire episodic memory at the low end.

To ratify the theory, a data-driven algorithm, called the FGP (Fetch, Generalise, Project) machine, was written. It can find high-end relationships between cases [7]. In one experiment, the machine is tested on a series of room descriptions encoded as simple attribute pairs (e.g. ((oven yes) (computer no) (coffee-machine yes) (sink yes) (sofa no))). The program is to describe the kind of room which has a particular attribute or attributes, and is given an appropriate probe vector (e.g. (oven yes)). The FGP machine first fetches all the cases which closely match the probe attributes. It places these into a "memory sandwich" and examines this longitudinally for common attributes. If all cases in the sandwich have a common attribute, the program generalises, guessing that all cases with the probe attribute will also have the common attribute. If many of the cases have some attribute, the program "projects", or speculates that this attribute could be characteristic, and recursively uses this attribute as a probe. If the memory sandwich returned by this recursive probe is a good match to the characteristic pattern of attributes built up so far, the program accepts the putative attribute; if not, the attribute is discarded and the next attribute considered. The process continues this fetch-generalise-project cycle until all attributes have been accounted for, returning a composite room description.

An FGP machine may be described formally as follows:

Let \( T \) be a feature tuple, \( \mathcal{M} \) be an unordered collection of feature tuples and \( \mathcal{L} \) be a list of \( \mathcal{M} \)'s, ordered by a suitably defined proximity metric to some arbitrary \( T \). Then let the following functions be defined:

- \( \text{fetch} (T, \mathcal{M}) \rightarrow \mathcal{L} \) Given a single pattern \( T \), returns an ordered list \( \mathcal{L} \) of patterns from \( \mathcal{M} \) which are closer than some threshold to \( T \) in the problem space.

- \( \text{generalise} (\mathcal{L}) \rightarrow T \) Given a list \( \mathcal{L} \) of patterns, generate a new pattern \( T \), which captures general features of all patterns in \( \mathcal{L} \). The contribution of each element of \( \mathcal{L} \) to the new generalised pattern depends on its ordinal position in \( \mathcal{L} \) and on the element's status as either a prototype or an ordinary case.

- \( \text{project} (T) \rightarrow T' \) Given a single pattern \( T \), returns a new pattern \( T' \), which contains a subset of the most "evocative" features of \( T \), and which thereby shifts subsequent processing into new regions of the problem place.

The aim is to answer a query input by the user. Queries can be either a pair \((T_0, a?)\) consisting of a test feature tuple and a single attribute, or else a single tuple \( T_0 \). An answer to the first query will be value of a for \( T_0 \), while the second query examines the cases for a prototypical redescription of \( T_0 \). In effect, it summons the memory walk process, asking "what does \( T_0 \) bring to mind?"