

The Copycat Project: A Model of Mental Fluidity and Analogy-Making

Tom Rochette <tom.rochette@coreteks.org>

November 2, 2024 — [36c8eb68](#)

0.1 Context

0.2 Learned in this study

0.3 Things to explore

1 Overview

2 Notes

2.1 Copycat and mental fluidity

2.1.1 Analogy Problems in the Copycat Domain

- Copycat’s architecture is neither symbolic nor connectionist
- Copycat architecture is an emergent architecture, in the sense that its top-level behavior emerges as a statistical consequence of myriad small computational actions, and the concepts that it uses in creating analogies can be considered to be a realization of “statistically emergent active symbols”

2.1.2 The intended universality of Copycat’s microdomain

- This project, which sprang out of two predecessors, Seek-Whence and Jumbo, has been under development since 1983
- The Copycat project is not about simulating analogy-making per se, but about simulating the very crux of human cognition: fluid concepts
- The reason the project focuses upon analogy-making is that analogy-making is perhaps the quintessential mental activity where fluidity of concepts is called for, and the reason the project restricts its modeling of analogy-making to a specific and very small domain is that doing so allows the general issues to be brought out in a very clear way

2.1.3 A perception-based, emergent architecture for mental fluidity

- One of the main ideas of the project is that even the most abstracted and sophisticated mental acts deeply resemble perception
- The inspiration for the architecture comes in part from a computer model of low-level and high-level auditory perception: the Hearsay II speech-understanding project

2.2 The Three Major Components of the Copycat Architecture

- The Slipnet: the site of all permanent Platonic concepts. It can be thought of, roughly, as Copycat’s long-term memory

- The Workspace: the locus of perceptual activity. It contains instances of various concepts from the Slipnet, combined into temporary perceptual structures. It can be thought of, roughly, as Copycat’s short-term memory or working memory, and resembles the global “blackboard” data-structure of Hearsay II
- The Coderack: can be thought of as a “stochastic waiting room”, in which small agents that wish to carry out tasks in the Workspace wait to be called

2.3 The Slipnet - Copycat’s network of Platonic concepts

- A network of interrelated concepts, each concept being represented by a node, and each conceptual relationship by a link having a numerical length, representing the “conceptual distance” between the two nodes involved
- There are roughly 60 concepts in Copycat’s Slipnet
- Conceptual links in the Slipnet adjust their lengths dynamically
- Conceptual distances gradually change under the influence of the evolving perception (or conception) of the situation at hand

2.3.1 Conceptual depth

- Each node in the Slipnet has one very important static feature called its conceptual depth
- This is a number intended to capture the generality and abstractness of the concept
- Assignment of conceptual depths amount to an a priori ranking of “best-bet” concepts

2.3.2 Activation flow and variable link-lengths

- Some details about the Slipnet’s dynamical properties:
 - there are a variety of link types, and for each given type, all links of that type share the same label
 - each label is itself a concept in the network
 - every link constantly adjusts its length according to the activation level of its label, with high activation giving rise to short links, low activation to long ones

2.3.3 Concepts as diffuse, overlapping clouds

- Although it is tempting to equate a concept with a pointlike node, a concept is better identified with this probabilistic “cloud” or halo centered on a node and extending outwards from it with increasing diffuseness
- Note that whereas the Slipnet changes over the course of a single run of Copycat, it does not retain changes from run to run, or create new permanent concepts
- The program starts out in the same initial state on every run
- Although the Slipnet responds sensitively to events in the Workspace by constantly changing both its “shape” and the activations of its nodes, its fundamental topology remains invariant

2.4 The Workspace - Copycat’s locus of perceptual activity

- At the start of a run, the Workspace is a collection of unconnected raw data representing the situation with which the program is faced

2.4.1 The constant fight for probabilistic attention

- The probability that an object will attract a prospective codelet’s attention is determined by the object’s salience, which is a function of both the object’s importance and its unhappiness

2.5 The Coderack - source of emergent pressures in Copycat

- All acts of describing, scanning, bonding, grouping, bridge-building, destruction, so forth in the Workspace are carried out by small, simple agents called codelets

- There are two types of codelets: scout codelets and effector codelets
 - A scout merely looks at a potential action and tries to estimate its promise
 - An effector actually creates (or destroys) some structure in the Workspace
- Each codelet, when created, is placed in the Coderack, which is a pool of codelets waiting to run, and is assigned an urgency value - a number that determines its probability of being selected from that pool as the next codelet to run
- Bottom-up codelets (or “noticers”) look around in an unfocused manner, open to what they find, whereas top-down codelets (or “seekers”) are on the lookout for a particular kind of phenomenon, such as success relations or sameness groups

2.5.1 Pressures determine the speeds of rival processes

- Any run starts with a standard initial population of bottom-up codelets (with preset urgencies) on the Coderack
- At each time step, one codelet is chosen to run and is removed from the current population on the Coderack

2.5.2 The shifting population of the Coderack

- Replenishment of the Coderack takes place constantly, and this happens in three ways:
 - Bottom-up codelets are continually being added to the Coderack
 - Codelets that run can, among other things, add one or more follow-up codelets to the Coderack before being removed
 - Active nodes in the Slipnet can add top-down codelets
- There is a feedback loop between perceptual activity and conceptual activity, with observations in the Workspace serving to activate concepts, and activated concepts in return biasing the directions in which perceptual processing tends to explore

2.6 The Emergence of Fluidity in the Copycat Architecture

2.6.1 Commingling pressures - the crux of fluidity

- Although some large-scale actions tend to look planned in advance, that appearance is illusory; patterns in the processing are all emergent

2.6.2 Time-evolving biases

- At the very start of a run, the Coderack contains exclusively bottom-up similarity-scanners, which represent no situation-specific pressures
- As these early codelets run, the Workspace starts to fill up with bonds and small groups and, in response to these discoveries, certain nodes in the Slipnet are activated
- In this way, situation-specific pressures are generated and cause top-down codelets to be spawned by concepts in the Slipnet
- As processing takes place and perceptual discoveries of all sorts are made, the system loses this naïve, open-minded quality, as indeed it ought to, and usually ends up being “closed-minded” - that is, strongly biased towards the pursuit of some initially unsuspected avenue

2.6.3 Temperature as a regulator of open-mindedness

- The more informed the system is, the more important it is that top-level decisions not be capriciously made
- There is a variable that monitors the stage of processing, and helps to convert the system from its initial largely bottom-up, open-minded mode to a largely top-down, closed-minded one
- This variable is given the name temperature
- What controls the temperature is the degree of perceived order in the Workspace

- Note that although the overall trend is for temperature to wind up low at the end of a run, a monotonic drop in temperature is not typical; often, the system's temperature goes up and down many times during a run, reflecting the system's uncertain advances and retreats as it builds and destroys structures in its attempts to home in on the best way to look at a situation
- What the temperature itself controls is the degree of randomness used in decision-making
- Note that the notion of temperature in Copycat differs from that in simulated annealing
- In simulated annealing, temperature is used exclusively as a top-down randomness-controlling factor, its value falling monotonically according to a predetermined, rigid "annealing schedule"
- By contrast, in Copycat, the value of the temperature reflects the current quality of the system's understanding, so that temperature acts as a feedback mechanism that determines the degree of randomness used by the system

2.7 The Intimate Relation between Randomness and Fluidity

3 See also

4 References