Notes on Hale (2010): What a Rational Parser Would Do

Karl DeVries

February 21, 2014

1 Introduction

GOAL: introduce "a broad, algorithmic-level framework for theories of human sentence comprehension."

Sentence processing is . . .

- *incremental*, i.e. partial representations a build up as words are encountered
- *predictive*, i.e. the parser has expectations (at some level of granularity) about upcoming material
- results in representations that match those posited by *competence* theories of the grammar

Components of rational analysis:

1. Precisely specify what the goals of the cognitive system are
2. Develop a formal model of the environment which the system is adapted to
3. Make the minimal assumptions about computational limitations
4. Derive the optimal behavioral function given (1-3)
5. Examine the empirical literature to see if the predictions of the behavioral function are confirmed
6. If the predictions are off, iterate

What are the goals of the system?

- The human sentence comprehension system's goals include *rapidly* arriving at the syntactic structure *intended* by the speaker.

What environment is the system adapted to?

- Human sentence comprehension is adapted to categorical and gradient information specified in the grammars of particular languages.

What are the computational limitations of the system?

Hale's answers:

- parsing is a search through a problem space
- transitions between states within the problem space result from the application of grammatical rules
- analysis is guided by nearness to task completion

Other possible answers:

- extremely limited amount of focal attention
- inability to rapidly recover information about serial order
- reliance on content addressable memory access mechanisms

Hale assumes a generalized left corner parser (GLC):

- Announce points are allowed to vary between rules
- In particular assume that adjuncts are bottom up while arguments are left corner

What should you do when you have more than one applicable rule, i.e. what state should you visit next if there are multiple options?

- work out a function $f(n)$ that tells you how 'good' a state is (conceptualized as nearness to the end state)
- to get A* search we set $f(n) = g(n) + h(n)$ where
  - $g(n)$ is the cost of the path from the initial state all the way to state n
  - $h(n)$ estimates the cost required to get from n to the goal state

2 A simple Model

Focus only on syntactic knowledge

Mixed parsing strategy

- Adjuncts are processed bottom up
- Everything else is processed from the left corner

Note: the parser optimizes the search function within the search space that's defined by the mixed parsing strategy. It does not create an optimal search space by exploiting an ability to tune the announce points of individual rules. What would it take to expand the model to allow this?

The search heuristic:

- Assume that shorter search paths are optimal
- Set $g(n)$ to the number of steps taken to reach a particular state
- Set $h(n)$ as an estimate of distance to completion

How to set $h(n)$:

- Simulate the parsing strategy on a large corpus
- Classify states in terms of what is on the stack i.e. what has been found and what is expected
- For each state find out how far (on average) it is from the goal state, S
- If you encounter a state you've never seen before, look at the longest sub-stack for which experience is available

How much effort goes into a single sentence?

- Search length measures search difficulty

(1) The horse raced past the barn fell — 115 states

(2) The horse raced past the barn — 43 states

3 Three Examples

3.1 Garden Pathing

(3) While Mary was mending(,) a sock fell on the floor

Experience based heuristic rates expanding including *a sock* as an argument of the VP as better, than projecting an intransitive VP for mending.

- 43 search nodes on the garden path, 38 nodes on the non-garden path.

This doesn't seem as impressive as the horse-race-fell case.

Note: we have to interpret these state searches in relative terms. The GP here is as easy as the horse-race-fell case.

Note: this model assumes monotonic structure building. No repair operation, instead we return to previous states. How does this square with the ERP lit which shows that people seem to be sensitive to not only the violation of an expectation but also the source of that expectation?

3.2 Attachment paradox

(4) a. I gave her earrings on her birthday
  b. I gave her earrings to another girl

The ambiguity above is unproblematic. In both searches, there is a mixture of correct and incorrect attachment of *her* and *earrings*

3.3 Local Coherence

(5) The coach smiled at **the player** [**tossed a Frisbee** by the opposing team]

Substrings that locally cohere are still globally inconsistent (even when only considering the initial string)

Key requirement of a model: it must go through some states where the locally coherent material is parsed together

The globally correct sentence requires a post modifier, which has to be build bottom up

This results in a three long stack: The [NP] prediction, NP (the player), RC (tossed a Frisbee)

This stack is very uncommon (18 cases) and on average 76 steps from completion

So, A* leads away from exploring states like this, and it projects an S above NP (the player)

Hale's Note: The right prediction depends on "the search procedure being free to explore locally coherent states...This outcome spotlights the trade-off between grammatical deduction and graded heuristic guidance. It may be that alternative behavioral tasks and individual differences can be modeled in terms of the trade-off between these two theoretical devices."

Note: This prediction also rests on the by fiat bottom up nature of adjunct parsing, which doesn't itself fall out of higher level properties of the system

4 The Refined Model

Simple model treats states as sequences of category in the stack

Do some sought categories (expectations) come with increased time to completion? Yes

Basic idea (= - the math):

- We take grammar rules to have certain wights (a PCFG)
- We figure out the uncertainty of a given non terminals expansion, by summing each expansion:

  hXP = sum([Prob(rule) * log2(Prob(rule)) for rule in XPrules])

- From there we can figure out the average uncertainty of any terminal

We implement this in A* search by penalizing complex categories.

Say that the entropy associated with PP is greater than that associated DP, the parser would have to peruse the DP hypothesis until it was no longer tenable before pursuing the PP hypothesis.

(6) a. The Australian woman saw the famous doctor had been drinking quite a lot
    b. Before the woman visited the famous doctor had been drinking quite a lot

The (b) cases go through fewer states, than the (a) cases, since the parser wants to avoid projecting another clause in (a)