Automaton Theories of Human Sentence Comprehension

John T. Hale (2014)

KU 건국대학교 KONKUK UNIV.

Brain & Cognition Research Center

발제자: 박기효

2020. 02. 29.

Chapter 6. Experience as a Control Strategy for Incremental Parsing

Introduction

- The heuristics from the Pereira-Shiber fomalization of Garden-Path Theory

→ Compatible with lots of human processing, especially on the garden-path sentences. But when applied more widely, to free text, their performance is disappointing.

- Franz (1996): No set of fixed heuristic pricinples performs well enough to be taken as a psycholinguistic law, or to help much in a natural-language computer system.

- It appeared that quite a lot of rules woud be needed to approximate the syntactic attachment preferenes of even a monolingual speaker.

→ Alternative views: To theorize not the attachment preferences themselves, but rather their relationship to experiences that a hearer in the relevant language community would have (Mitchell et al, 1995; Jurafsky 1996)

Introduction

- Two examples of how learning might fit into incremental parsing

1. Reinforcement Learning
   - A new model whose syntactic attachment preferences do in fact change with experiences.

2. Batch approach
   - Experience with a text corpus is summarized in large table.
   - Alleviate the restriction to single-path parsing to show how the *appearance* of parallel processing might emerge as a result of experience.

Reinforcement-learning

- Figure 26

- Considering each of the rules of this grammar, we adopt the parsing strategy, i.e., the left-corner parsing.

  → They are announced as soon as the first daughter symbols has been found bottom-up.

- In these automaton models, ambiguity becomes nondeterminism about selecting the next parser ation.

phrase structure rule			comment
S1	→	S	dummy root rule
S	→	NP VP	subject-verb sentence
NP	→	DT NN	
NP	→	NP VP	reduced relative as verb phrase adjunction rule
DT	→	the	
NN	→	barn	
NN	→	horse	
VP	→	VBN PP	past participle with locative
VP	→	VBD PP	past tense with locative
VP	→	VBD	past tense intransitive
VBN	→	raced	ambiguous!
VBD	→	raced	
VBD	→	fell	
PP	→	IN NP	
IN	→	past	

FIGURE 26 Grammar for MV/RR garden-path

Reinforcement-learning

- The correct sequence of actions is congruous with a traversal of the globally-correct tree. (Fig. 28)

- Sine qua non: The odds of hitting upon exactly this sequence are not good.

shifting the
projecting DT→the
projecting NP→DT-NN
shifting horse
projecting NN→horse
matching NN
projecting S→NP-VP
shifting raced
projecting VBN→raced
projecting VP→VBN-PP
shifting past
projecting IN→past
projecting PP→IN-NP
shifting the
projecting DT→the
projecting NP→DT-NN
shifting barn
projecting NN→barn
matching NN
shifting fell
projecting VBD→fell
projecting VP→VBD
projecting NP→NP-VP
matching NP
matching PP
matching VP
projecting S1→S
matching S1

FIGURE 28 Sequence of operators leading to the globally-correct analysis in MV/RR example

Reinforcement-learning

- These bad odds reflect the preponderance of choice points (Fig. 29)

- A Soar model destructively updates the contents of working memory, maintain exactly one stack of sought/found grammar symbols and basing its decisions on whatever knowledge is available, but there is no knowledge available so Soar is indifferent.

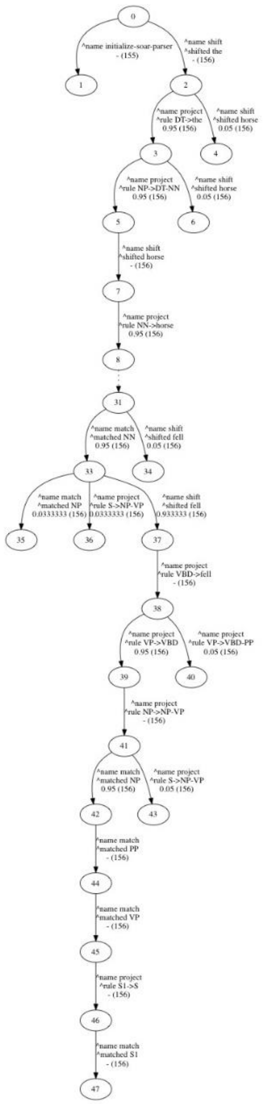


FIGURE 29 Search space in the MV/RR ambiguity after considerable reinforcement learning. The dotted line omits nodes 9 through 30.

Reinforcement-learning

- We can define a schedule of rewards and punishments for Soar's built-in RL facility (Tab. 14)

- Positive numbers: reward ↔ negative ones: punishments

- This feedback is meted out only at the end, but via Soar's built-in RL facilities they come to tune the choice of all operators seleted throughout the analysis process.

- At the end of a failed run, the model goes back to the beginning of the sentence and tries again.

- Each choicepoint is a tie impasse whose resolution gets rcecorded in a chunck, which express preferences.

$+1$	all words consumed, stack is empty
-1	stack is still occupied but no more words left
-1	blocked; no actions are applicable

TABLE 14 Simple reward schedule for pushdown automata

Reinforcement-learning

- In this particular model, these chunks have a numerical strength parameter: Q-learning

- Q-learning iteratively estimates the expected reward that would come from taking an action in a state.

- Since reward in this formulation comes only at the end, the model learns backwards: first it straightenes out the the last choice point, then the next-to-last, the next-to-next-to-last, et cetra.

- Although the model does learn, when extended to larger grammars, this method seems unrealisitically slow probably due to the over-simple notion of feedback (just) at the end.

- It remains an open research question how best to formulate a notion of feedback that leads to preferring one syntactic structure over another.

Informed Serach with a Distance Estimate

- There's a distance factor scaling the +1s and -1s at the end of candidate state trajectories.

- If any finished state is acceptable, then a rational parser would move along the path whose estimated distance to a finished state is shortest.

- **Informed Serach**

- An estimate of the distance to the goal informs a searcher about which state to choose.
  - Estimate: In parsing, ditances cannot be known with certainty. → CORPUS (Tab. 15)

- Rather than iteratively estimating expected reward, by acquiring tables like tab. 15, estimate can be looked-up and associated with search states.

stack contents	attestations	average # steps to goal	standard error
[VP] S	55790	44.936	0.1572
S	53991	10.542	0.0986
[NP] S	43635	33.092	0.1633
NP	38844	55.791	0.2126
NP [S] S	34415	47.132	0.2122
[S] S	33578	52.800	0.2195
[PP] S	30693	34.454	0.1915
IN [PP] S	27272	32.379	0.2031
DT [NP] S	22375	34.478	0.2306
[AUX] [VP] S	16447	46.536	0.2863
VBD [VP] S	16224	43.057	0.2826
VB [VP] S	13548	40.404	0.3074
the [NP] S	12507	34.120	0.3046
NP [NP] S	12092	43.821	0.3269
DT	10440	66.452	0.3907

TABLE 15 Stack configurations attested in the Brown corpus

The A* Formulation and Multipath Parsing

- In a sigle-path parser like the Soar models, it ultimately results in *one* operator being applied.

- A multipath arrangement maintains the fringe of the serach tree, the set of parser states that have been reached but whose successors have not yet been explored.

- A parser state that is farther along in the sentence will presumably have a smaller heuristic distance estimate than a state that is less far along.

1). To make states comparable along this dimension, each search state n receives a cost estimate, $\hat{f}(n)$ that includes both the number of states that have been taken to get to n, as well as an estimate ($\hat{}$) of how many steps it will take to get from n to a goal state.

$$\hat{f}(n) = g(n) + \hat{h}(n) \tag{6.1}$$

- A* algorithm offers a gurantee that the shortest path will be found in the fewest number of steps.
  □ 'Admissibility' that amounts to a promise never to underestimate the true distance.

The A* Formulation and Multipath Parsing

- Two interesting points of the A* Formulation
    1. A shorter, simpler analysis being preferable.
    2. It saves the cognitive modeler from having to separately specify the exact number of paths that are pursued simultaneously in multipath parsing, i.e. to be determined indirectly by the heuristic.

- If the heuristic is very “sharp”, then A* will behave more like depth-first serach (single-path)

- On the other hand, if the heuristic is not so sharp, then A* will degenerate into breadth-first serach (multi-path)

Informed Serach Emulating Late Closure

"While Mary was mending a sock fell on the floor"

- Figure 30 (right next) shows the entire serach tree.
  - □ Heavy lines: glollay-correct pathway
  - □ Doted lines: the garden-path analysis

- In Figure 31, the essential point is that prior experience makes the garden-path more attractive.

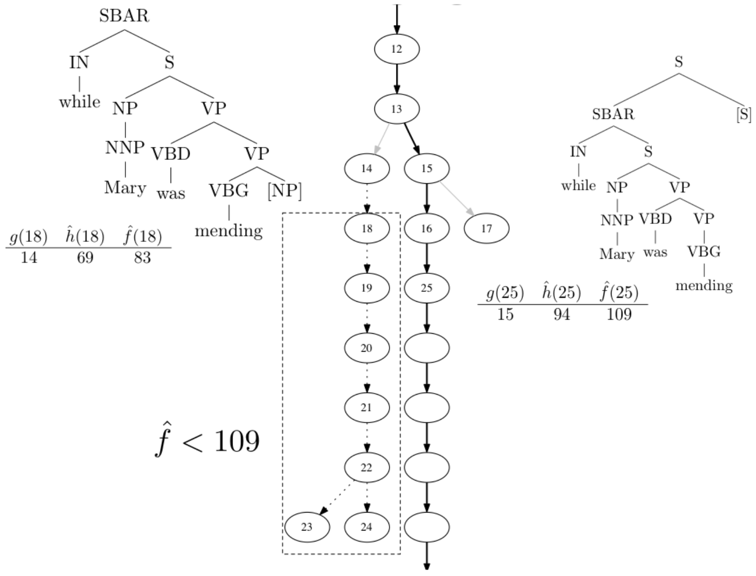


FIGURE 31 The motive for the garden path. This Figure shows a subset of the nodes in Figure 30.

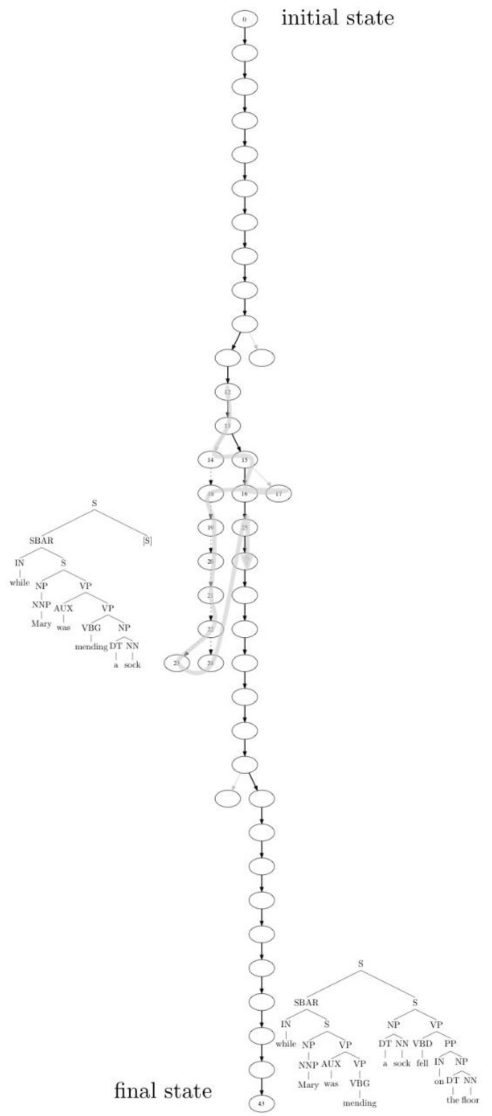


FIGURE 30 Garden-path'd by estimates from the Brown corpus

Informed Serach and the Appearance of Parallel Processing

- In cases where the heuristic is not so sure what to do, exploration begins to look more and more like breadth-first search.

- Where differenced in hat f are small, informed serach can derive a kind of parallel processing even though it is extending just one state at a time.

- It does so by flitting back and both between different analyses.

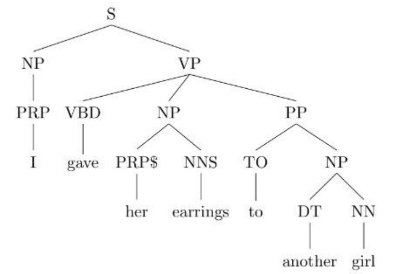
- This is one way that informed search could explain the absence of garden path effects in sentences where fixed heuristics would imply that they should exist.

Informed Serach and the Appearance of Parallel Processing

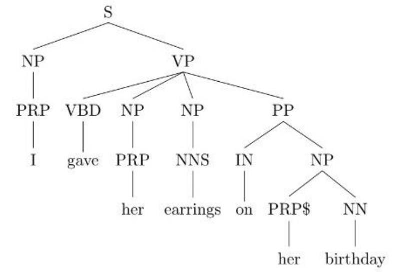
- Figrue 32.
  → No matter which was fixed preferences are formulated, they derive a prediction of garden-pathing on half the items.

- One way out of the dilemma is to say that both analysises are built in parallel (Fig. 33)

- This interleaving mimics (apparent) parallel parsing by switching back and forth between alternative analysis paths.

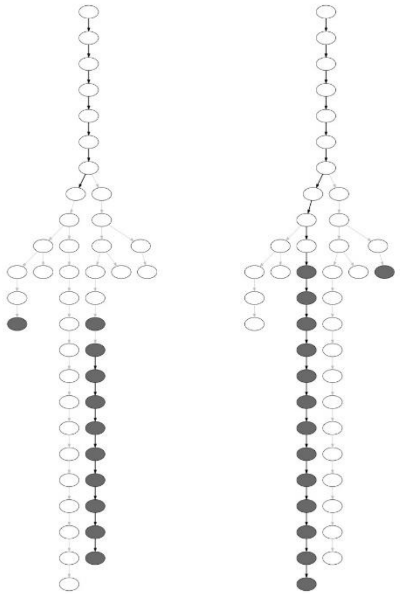


(a) Analysis of 6 respects Late Closure; "earrings" is part of the most recent phrase, NP



(b) Analysis of 6 conflicts with Late Closure; "earrings" could have been part of the most recent phrase, NP

FIGURE 32 Attachment paradox for fixed heuristics



(a) 6 globally incorrect possessive NP shaded

(b) 6 globally incorrect pronominal NP shaded

FIGURE 33 Parallel parsing as time-sharing: darkened circles indicate states in which the 'other' interpretation is being explored.

감사합니다!