Automaton Theories of Human Sentence Comprehension

John T. Hale (2014)

발제자: 박기효

2020. 03. 15.

Chapter 8-9. Surpisal and Chunking + Conclusion

Introduction

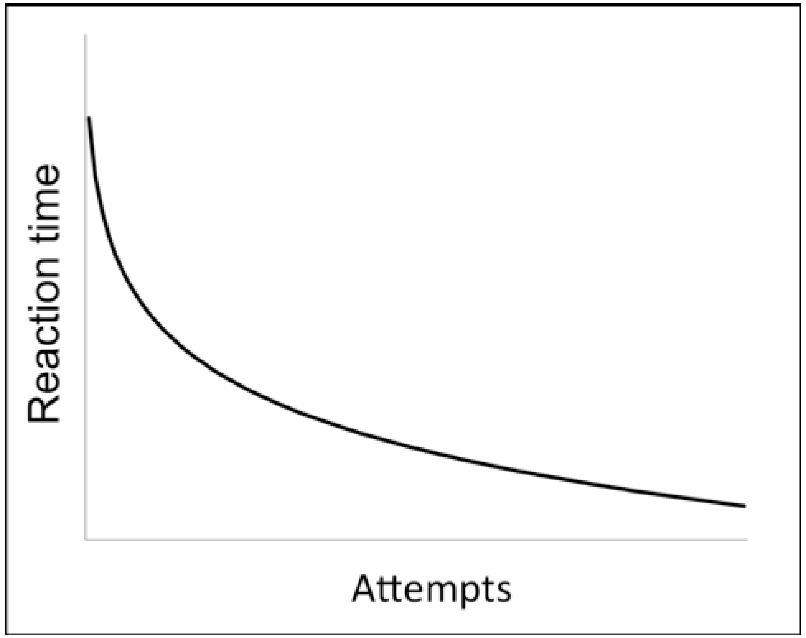
- Chunking Theory of Learning (CTL)

1. Chuncking to account for the power law of practice – the logarithm of the reaction time for a particular task decreases linearly with the logarithm of the number of practice trials taken.

2. Language use as a highly-practiced skill
   → Ubiquitous frequency effects at all scales

The combination of CTL and Generalized Left-Corner pasing
→ To simulataneosurly explain:

   i. How parsing happens.
   ii. How it becomes frequency-sensitive.
   iii. What its relationship to grammar is.



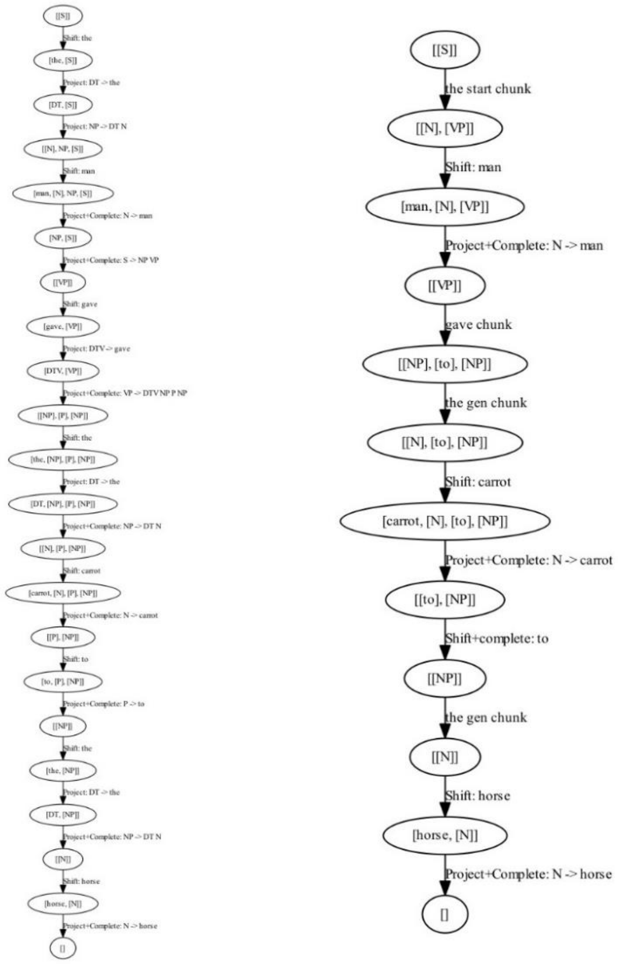
How chunking applies to parsing

- The CTL's basic idea is that subtasks of a cognitive process may be combined, yielding a new operator that excutes more quickly than its components (Figure 36)

- This sense of chuncking is slightly different than the one in ch. 5, where chunkcs recorded the outcome of decisions that were made at choicepoints.

- In fig. 36, rather each chunk presents a new macro-operator that combines several old operators into a single new one.

- With these macro-operators now available, the search space is coarser (36(b)) than it was before (36(a)).



(a) Without chunking (b) With chunking

FIGURE 36 Left-corner parsing without and with chunking. Search spaces for "the man gave the carrot to the horse"

Identifying good chunks

- The example chunks in fig. 36 were motivated by intuitive notions of predictability.

- We can standardize the identification of good chunks by viewing a Treebank as a stream of parser actions.

- In table 16, **'cohesion'** give a log-likelihood ratio between the value that should have been expected if each of the three parsing actions were (probabilistically) independent, compared to the observed number of attestations of this triple.

- A parser-action tiple with high cohesion is likely to become a macro operator under CTL.

- Same results in French (Table. 17).

chunk	cohesion
shift preposition ; project PP $\rightarrow$ P NP; shift determiner	472371.7530
⋮	
shift determiner ; project NP $\rightarrow$ Det N ; complete N	349337.5764
⋮	
project NP $\rightarrow$ Det Adj N ; complete Adj ; complete N	144445.6793
⋮	
project NP $\rightarrow$ Adj ; shift colon ; shift verb	10.0435

TABLE 16 Example chunks from an English Treebank

Reading faster when the chunk is familiar

- The CTL explains speedup due to practice in terms of new mental representations:

  Performance Assumption
  The performance program of the system is coded in terms of high-level chunks, with the time to process a chunk being less than the time to process its constituent chunks.

- While the exact threshold between having a chunk and not-having chunk will depend on the details of a particular algorithm, what is clear is that parser-action triples with higher cohesion values are more likely to get chunked.

- If a human reader is using the chunks rather than the individual component GLC parsing actions, it should be read faster under the CTL performance assumption.

- To evaluate this idea, we used linear regression to predict eye-fixation durations in eyetracking copora.

English & French study

	Estimate	Std. Error	t-value	sig? at $p < 0.01$?
(Intercept)	233.4317	0.2614	893.05	
cohesion	-6.0852	0.2614	-23.28	yes

TABLE 18 English: as only predictor

	Estimate	Std. Error	t-value	sig at $p < 0.01$?
(Intercept)	.02239	2.899	77.223	
cohesion	-5.466×10^{-4}	7.762×10^{-5}	-7.042	yes

TABLE 20 French: as only predictor

	Estimate	Std. Error	t-value	sig at $p < 0.01$?
(Intercept)	5.2807527	0.0003774	13993.605	
number of chars	0.0083617	0.0002149	38.905	yes
word prob	-0.0136493	0.0002259	-60.433	yes
cohesion	-0.0010350	0.0003840	-2.695	yes

TABLE 19 English: with other predictors

	Estimate	Std. Error	t-value	sig at $p < 0.01$?
(Intercept)	-0.2155	3.338	-6.457	
cohesion	-1.111×10^{-4}	4.365×10^{-5}	-2.545	$p = 0.0109$
number of chars	21.22	0.4912	43.198	yes
word prob	16.08	0.5794	27.749	yes

TABLE 21 French: with other predictors

Discussion

- These two studies lend support to the idea of chunks as psychologically real sequences of parser actions that make reading go faster.

- Via the CTL, one can say that constructions get stronger with usage, while at the same time interpreting them with respect to particular rules of a generative grammar.

- An important tradition in computational linguistics has considered larger chunks of phrase structure as grammatical entities.

- The approach taken here, inspired by the CTL, is different in that the chunking happens at the level of parser actions rather than the grammar.

Discussion

- Although the results are promising, many questions remain open. One is mathematical in nature:

How suprsials change
For a given probabilistic grammar *G,* choose a canonical transistion machine *M(G)* that assigns the same probabilities as *G* does to its language, *L(G):*

1. How do the surprisal values assignend to sentences in *L(G)* change with the addition, via chunking, of additional transitions summarizing sequences of previously-extant transitions?
2. How should renomalization take place to ensure that the probability model remains well-defined?
3. How quickly does the distribution assigned by the transition-based representation start to diverge from that assigned by *G*?

Discussion

- Another more developmental question has to do with:

Lifetime exposure

- If language comprehension is really a procedural skill that gets tuned by use, then we should expect that the number of times a person has heard a construction in a given context should predict the amount of time it takes him or her to comprehend.

- Evaluate this claim using longitudinal data to estimate as precisely as possible the number of times that individual children have heard particular constructions that are within, or proximal to their linguistic competence.

Conclusion (Ch. 9)

- The automaton view laid out in this book is menat to bridge tha gap: How do grammar and parser interrelate?

- The proposed answer is that a grammar rule should label the name of a structure-building action.

- The automaton's transition causally does something, whereas the grammar rule interprets the linguistic significance of that action.

- Remainig (particular) questions:

Conclusion (Ch. 9)

- Remainig (particular) questions:

1. Which grammar is the right one?

- One might imagine that data on language behavior could reach back up through a formalized theory of performance to help linguists decide which grammar is the right one.

- One some level this is true, but it also invites a category mistake.

- The empirical finding that a parituclar construction is easy rather than hard does not speak to the raw possibility of that construction in a language, which does not speak to the relationship between the construction and the its typological cousins on other languages.

- The true bite of the automaton view is the proposal that individual rules are the right granularity for processor operators, which is an attractive opening bid as it keeps things simple.

Conclusion (Ch. 9)

- Remainig (particular) questions:

2. Neural realization

- What does it mean for a grammatical relation to be “tokened” in the mind?

- What brain-state transitions are characteristic of a comprehension episode?

- These questions, about the physical basis of syntactic distintions, are analogous to phonetic questions about the physical basis of speech sounds.

감사합니다!