We have seen how mental states and cued associations can define complex ideas.

This lecture will describe how these complex ideas can be encoded and decoded into sentences.

### Contents

- **11.1 Linguistic knowledge** ........................................... 1
- **11.2 A lexicon of typed sign states** ................................. 2
- **11.3 A grammar of prediction rules** ................................. 3
- **11.4 Left-corner processing operations** ............................. 4
- **11.5 Example decoding** .................................................. 6

### 11.1 Linguistic knowledge

Each language defines a **lexicon** of basic **signs** and a **grammar** of rules for composing them.

A sign is a mental state (elementary predication) for a linguistic event (e.g. a use of a word), with:

- a type (‘0’ association), like any elementary predication, but specifically consisting of:
  - a **category** (e.g. transitive verb, intransitive verb, etc.) in a given domain $C$,
  - a **form** (e.g. ‘borrow’);
- a set of participants (numbered associations ‘1,’ ‘2,’ ...), like any elementary prediction, but specifically defining **syntactic argument referents**.
- a new association for a **referent** (‘ref’);

Signs may have **simple/primitive categories** (with no missing arguments) in domain $P \subset C$: 
• category ‘V’: inflected verb expression (‘[V she knew the secrets]’)
• category ‘B’: base-form verb expression (‘they required that [B she know the secrets]’)
• category ‘I’: infinitive verb expression (‘they wanted [I him to know the secrets]’)
• category ‘G’: gerund verb expression (‘they feared [G him knowing the secrets]’)
• category ‘A’: adjective expression (‘they considered [A him knowledgeable]’)
• category ‘N’: noun expression (‘they deplored [N his knowledge of the secrets]’)
• category ‘D’: determiner expression (‘they deplored [D the knowledge’s] effect’)
• ...

Signs may also have complex categories (missing syntactic arguments of various categories):
  • γ-a δ: a sign lacking an argument of category δ ahead of it (‘[N-aD toy]’ lacks a determiner),
  • γ-b δ: a sign lacking an argument of category δ behind it (‘[N-b(N-aD) some]’ lacks a noun),
  • ...

We may think of the various argument categories -a δ, -b δ as separate features of a sign’s type.
These categories resemble those in a categorial grammar [Ajdukiewicz, 1935, Bar-Hillel, 1953].

11.2 A lexicon of typed sign states

Exposure to stimulus w (e.g. ‘borrow’) constructs a new mental state a for a sign of category γ.
• A state can be ‘new’ if it has some unique (‘temporal’) features, but familiar type features.
  (This is not language-specific — anything you experience constructs a new state.)

Category features γ are learned with activation proportional to the probability of word w given category γ.
The sign state resulting from stimulus w then has a mixture of activation for all possible categories γ.
Sign types \((V\cdot a\cdot N\cdot b\cdot N: \text{borrow})\) produce learned (generalized) inferences of cued associations, as needed:

\[\text{N-b(N-aD):most} \quad \text{N-aD:people} \quad \text{V-aN-bN:borrow}\]

\[\text{state for instance of word ‘borrow’}\]

\[\text{state for things borrowed}\]

\[\text{11.3 A grammar of prediction rules}\]

A grammar of learned (generalized) rules constructs predicted signs \(b\) given antecedent signs \(a\).

This construction is learned with activation depending on the type and cued associations of \(a\) and \(b\).

The resulting state is therefore also a mixture of activation for all predicted sign states \(b\).

- Argument attachment: stores cued associations to referent of argument (nuclear scope)

  Initial argument attachment (where each \(\varphi_n \in \{-a,-b\} \times C; b\) has no cued association \(\geq N\)):

  \[
  \gamma \\
  \text{(e.g. N)} \\
  \Rightarrow \\
  \gamma \\
  \text{(e.g. N)} \\
  \text{p}\varphi_1...\varphi_{N-1}\cdot ay \\
  \text{(e.g. V-aN)}
  \]

  Final argument attachment (where each \(\varphi_n \in \{-a,-b\} \times C; a\) has no cued association \(\leq N\)):

  \[
  \text{p}\varphi_1...\varphi_{N-1}\cdot by \\
  \text{(e.g. V-aN-bN)} \\
  \Rightarrow \\
  \text{p}\varphi_1...\varphi_{N-1}\cdot by \\
  \text{(e.g. V-aN-bN)} \\
  \gamma \\
  \text{(e.g. N)}
  \]
Modifier attachment: stores cued associations to restrictor of modificand (*all red cars rule*)

Initial modifier attachment (*a* has no cued association 1):

Final modifier attachment (*b* has no cued association 1):

These rules may be expressed as a **categorial grammar** [Ajdukiewicz, 1935, Bar-Hillel, 1953]. Predicting referential states together with signs like this is called **compositional semantics**.

### 11.4 Left-corner processing operations


Sentence processing maintains a sequence of sign fragments *a/b* with a designated current sign *b*:

Complex ideas can now be assembled by connecting lexical signs and composition rules:

- Frontier phase (add lexical signs and connect to existing sign fragments, or don’t):
Yes-match option:

![Yes-match option diagram]

No-match option:

![No-match option diagram]

- Junction phase (apply prediction rules and connect resulting sign fragments, or don’t):

  No-match option:

  ![No-match option diagram with junction phase]

  Yes-match option:

  ![Yes-match option diagram with junction phase]
11.5 Example decoding

From context, fork observed word *most*, predict argument, but don’t join to previous sign fragment:

Other predictions and expectations might also be superposed at $a_2$.

Then match *people*, predict $v_2$ as first argument (subject) of $a_3$, and don’t join:
Then fork *borrow*, predict second argument (direct object), and join to previous sign fragment:

Then fork *some*, predict argument, and join to previous sign fragment:
Finally, match *toys*, and join complete sign $a_3$ to previous sign fragment:

![Diagram showing the matching of signs and the joining of complete signs](image)

The structure of rule applications over time still looks tree-like.

**References**


