Language processing may be based on domain-general complex event prediction. This uses memory and generalization (learning) to recognize complex events (plans).

(Recall that events may be represented in the brain as elementary predications. We will assume events are also connected by elementary predications of causation.)

### 7.1 Complex events

Events can contain hierarchies of subevents, especially complex plans (complex ideas):

- `ants-in-mouth`
- `ants-on-stick`
- `ants-in-anthill`
- `stick-in-mouth`
- `stick-in-anthill`
- `stick-in-fingers`
- `fingers-in-anthill`
- `branch-in-fingers`
- `twig-away-from-branch`
- `hand-on-twig`
- `hand-away-from-branch`

Sub-events are related to parent events by ‘cause’ elementary predications.

When similar (recognition) operations are nested inside other operations, a process is called **recursive**.
7.2 Recognition of complex events using event fragments

Humans and (some) animals can recognize and re-create complex hierarchic events.
[Fuster, 1990, Botvinick, 2007]

Partial sequences of events can be grouped and stored as **event fragments** $a/b$, where:

- $a$ is a **predicted whole** top-level event or sub-event,
- $b$ is an **expected part** sub-event or observed event yet to come, which completes the whole.

E.g. **ants-in-anthill** can be accounted as **ants-on-stick/stick-in-anthill**.

Use cued association (‘$A$’) to directly link an individual expectation $b$ to a supported prediction $a$.

Near-complete sub-events can be chained together to save memory:
E.g. **ants-on-stick/stick-in-anthill** and **stick-in-fingers** form **ants-on-stick/fingers-in-anthill**.

When a recent event fragment is completed, it can be added to an earlier event fragment.
E.g. if **stick-in-fingers** is complete, it can satisfy **stick-in-anthill** with **fingers-in-anthill** expected.

Use cued association (‘$B$’) to directly link an individual prediction $a$ to a preceding expectation $b$.

Uncertainty about events may be modeled using superposed activation vectors, described earlier.

7.3 Recognition Model

This model maintains a sequence of event fragments accessible from the current expectation $b$:

E.g. $a'/b'$ is **ants-on-stick/stick-in-anthill**, $a/b$ is **stick-in-fingers/twig-away-from-branch**.

Crucially, this store can only be a few elements long before interference causes trouble.

The model also assumes a set of learned **prediction rules**:
E.g. ants-on-stick \((a)\) is composed of ants-in-anthill \((c)\) followed by stick-in-anthill \((b)\).

Here, \(a\), \(b\), and \(c\) might be connected by a ‘cause’ elementary predication (magenta lines).

Complex ideas can now be assembled by connecting observed events to event fragments...

- ‘Fork’ decision (add observed event and connect to existing event fragment, or don’t):
  
  No-fork outcome (set current prediction):

  ![Diagram of no-fork outcome]

  Yes-fork outcome (check types, store cued association from \(a'\) to \(b\), set current prediction):

  ![Diagram of yes-fork outcome]

- ‘Join’ decision (apply prediction rule and connect resulting event fragment, or don’t):
  
  No-join outcome (apply rule, store cued association from \(b'\) to \(a'\) and \(a'\) to \(b\)):

  ![Diagram of no-join outcome]

  Yes-join outcome (check types, apply rule, store cued association from \(b'\) to \(a\)):

  ![Diagram of yes-join outcome]

Fork and join can be implemented in simple neural networks, generalized by procedural learning.

These operations can recognize any branching event structure using a minimum amount of memory.
7.4 Example recognition by hierarchic sequential prediction

Here is an example of recognizing a complex plan from observations.

Start with observation of anthill $c_1$, predict ants on stick $a_1$, and expect a stick in the anthill $b_1$:

Perhaps other predictions and expectations, like pushing over the anthill, are also superposed at $b_1$.

Then fork observation of stick $c_2$, predict twig $a_2$, but don’t join, leaving a new event fragment:

Then fork off observation about grabbing twig $c_3$, and join it to the previous event fragment:

Then don’t fork observation $c_4$ to complete $a_2$, and join it to the previous event fragment, leaving
only one event fragment:

Then don’t fork fingers at anthill \( c_5 \), predict ants in mouth \( a_5 \), expect stick in mouth \( b_5 \):

The structure of rule applications over time can be drawn as a tree:

### 7.5 Practice

Assume the following complex event is being recognized:
and the following event fragments exist after the observation of hand-to-rock:

Draw the event fragments that would exist after the observation of stone-in-hand.
Which operations (yes-fork, no-fork, yes-join and no-join) are used at this observation?

References

