Computational Cognitive Science 2026/27

Week 9

Frank Keller

Tutorial 6: Surprisal and Sentence Processing

This tutorial draws mainly on Lecture 16 (sentence processing), building on the idea of information/surprisal from Lecture 13. We will use word probabilities from a pretrained language model to make predictions about reading difficulty, then compare those predictions with **published human eye-tracking data**. We will not train a neural language model or implement a syntactic parser.

Part 3 uses a CSV file of **genuine GPT-2 probabilities** `gpt2_relative_clause_probabilities.csv`, which is included in this zip file. The probabilities were generated in advance for you. However, if you're curious about how this is done, have a look at `generate_gpt2_probabilities.py`.

Part 1: From word probabilities to surprisal (10 minutes)

Sentence processing is *incremental*: as each word arrives, a reader updates their expectations. A word's **surprisal** is

$$S(w_k) = -\log_2 P(w_k \mid w_1, \ldots, w_{k-1}).$$

The surprisal hypothesis predicts that words with higher surprisal will, *other things being equal*, be harder to process.

Consider the prefix *She spread the warm bread with ...*. Suppose a model assigns the following conditional probabilities to some possible next words. These alternatives do **not** exhaust all possible continuations.

```
bread <- data.frame(
  word = c("butter", "jam", "cheese", "socks"),
  probability = c(0.50, 0.25, 0.125, 0.001)
)
print(bread)
```

```
##     word probability
## 1 butter       0.500
## 2    jam       0.250
## 3 cheese       0.125
## 4  socks       0.001
```

Question 1 Write an R function `surprisal()` that converts probabilities to surprisal in **bits**. Apply it to all four words and add the resulting column to `bread`. Which word should be the most difficult to read, according to the surprisal hypothesis? What happens to surprisal when the probability is halved?

```
surprisal <- function(p) {
  # Replace this line. Hint: log2(p)
  NA_real_
}
```

```r
bread$bits <- surprisal(bread$probability)
print(bread)
```

```
##     word probability bits
## 1 butter       0.500   NA
## 2    jam       0.250   NA
## 3 cheese       0.125   NA
## 4  socks       0.001   NA
```

Question 2 Suppose the context includes *either*, as in *Peter read either a book or . . .*, making *an* after *or* more likely. In a simplified illustration, assume the probability of *an* rises from 0.10 to 0.40. Calculate its surprisal in each context. Predict the **direction** of the reading-time effect. (These probabilities are hypothetical, not estimates from the experiment by (Staub and Clifton 2006).)

```r
p_without_either <- 0.10
p_with_either <- 0.40
# Your code here
```

Part 2: Examining human reading times (20 minutes)

In Lecture 16 we compared **subject relative clauses (SRC)** and **object relative clauses (ORC)**:

- SRC: *The reporter that attacked the senator admitted the error.*
- ORC: *The reporter that the senator attacked admitted the error.*

Object relative clauses are generally more difficult to process. Where does this additional difficulty occur?

We will use **actual published condition means** from (Staub 2010) (Experiment 1, Table 1). Participants' eye movements were recorded while they read sentences with SRCs or ORCs. The measures below are means across observations, **not individual-participant data**:

- **Gaze duration**: summed initial fixations on a word before the eyes leave it.
- **Go-past time**: time from first entering a word until moving past it to the right, including time spent re-reading earlier words.

For each condition we list regions by grammatical function, rather than by their linear position. In an SRC, the relative-clause verb precedes the determiner and noun; in an ORC it follows them. Therefore the rows are **not** a word-by-word time series, and regions such as the noun contain different words across conditions.

```r
# Staub (2010), Experiment 1, Table 1. Means in milliseconds.
regions <- c("that", "determiner", "noun", "RC verb", "main verb")
reading <- data.frame(
  region = regions,
  src_gaze = c(237, 239, 316, 270, 357),
  orc_gaze = c(250, 249, 266, 318, 346),
  src_gopast = c(283, 272, 375, 333, 429),
  orc_gopast = c(303, 382, 459, 420, 432)
)
print(reading)
```

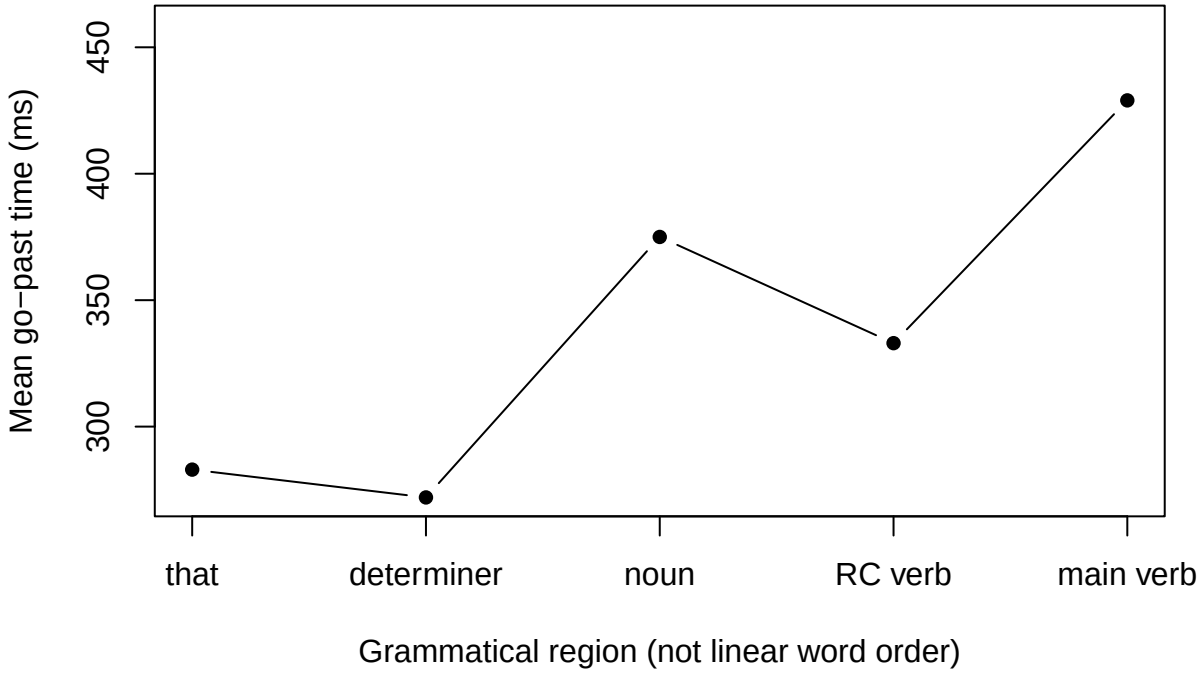
```
##       region src_gaze orc_gaze src_gopast orc_gopast
## 1       that      237      250        283        303
## 2 determiner      239      249        272        382
## 3       noun      316      266        375        459
## 4    RC verb      270      318        333        420
## 5  main verb      357      346        429        432
```

Question 3 Calculate the difference **ORC minus SRC** at each region, separately for gaze duration and go-past time. Display both differences alongside the region names. Which regions show the largest costs for ORCs, and does the answer depend on the eye-movement measure?

```
# Your code here; for example:
# reading$diff_gaze <- reading$orc_gaze - reading$src_gaze
```

Question 4 Make a plot of **go-past time** against region, with separate lines for SRCs and ORCs. You can use `plot()` and `lines()` as shown below. Add a legend. Based on this plot, at which regions does ORC processing appear more difficult? Note that the x-axis orders regions for comparison, **not** in the order in which they are read.

```
x <- seq_along(reading$region)
plot(x, reading$src_gopast, type = "b", pch = 16,
     xaxt = "n", ylim = range(reading[, c("src_gopast", "orc_gopast")]),
     xlab = "Grammatical region (not linear word order)",
     ylab = "Mean go-past time (ms)")
axis(1, at = x, labels = reading$region)
```




```
# Add the ORC line and a legend below:
# lines(x, reading$orc_gopast, type = "b", pch = 17, lty = 2)
# legend("topleft", c("SRC", "ORC"), pch = c(16, 17),
#        lty = c(1, 2), bty = "n")
```

Question 5 Compare the noun and RC-verb regions. How can the difference between **gaze duration** and **go-past time** inform us about the contribution of re-reading/regressions? Why should we avoid treating these five published means as though they were five independent participants in a statistical significance test?

Part 3: Testing an expectation-based explanation (15 minutes)

Surprisal is a *linking hypothesis*: a probabilistic model predicts which words are unexpected, and this is used to predict human processing difficulty. We will now use **actual conditional word probabilities from GPT-2**, a pretrained autoregressive language model (Radford et al. 2019).

The model scores the same target word in each of the two sentence contexts:

- SRC: *The reporter that attacked the senator admitted the error.*
- ORC: *The reporter that the senator attacked admitted the error.*

For instance, the ORC probability of *attacked* is $P(\text{attacked} \mid \text{The reporter that the senator})$. GPT-2 may split words into several tokens; the supplied probabilities multiply the conditional probabilities of **all tokens in each target word**. These are real model outputs, not hand-chosen numbers.

Important: These probabilities come from **one illustrative sentence pair**, whereas the reading times from (Staub 2010) in Part 2 are means over many experimental items. They allow an exploratory comparison, **not a direct quantitative evaluation** of GPT-2 on the stimuli used by (Staub 2010). Also, we compare regions by grammatical function, not their reading order.

```
# The CSV is produced by generate_gpt2_probabilities.py (instructor setup).
# No Python or language-model packages are needed for this R analysis.
model_file <- "gpt2_relative_clause_probabilities.csv"
if (!file.exists(model_file)) {
  stop(paste("Missing", model_file,
             "- ask your tutor for the GPT-2 predictions CSV."))
}
expectations <- read.csv(model_file, stringsAsFactors = FALSE)
expectations <- expectations[match(regions, expectations$region), ]
stopifnot(!anyNA(expectations$region),
          all(expectations$src_prob > 0 & expectations$src_prob <= 1),
          all(expectations$orc_prob > 0 & expectations$orc_prob <= 1))
print(expectations[, c("region", "word", "src_prob", "orc_prob")],
      row.names = FALSE)
```

```
##      region     word     src_prob     orc_prob
##        that     that 0.0021913142 0.0021913142
##  determiner      the 0.1131102965 0.0230793237
##        noun  senator 0.0020720549 0.0016407099
##     RC verb attacked 0.0016569279 0.0006965855
##   main verb admitted 0.0003000433 0.0001159790
```

Question 6 Use your `surprisal()` function to calculate SRC and ORC surprisal for every region. Then calculate `diff_bits`, the ORC-minus-SRC surprisal difference. At which regions does GPT-2 predict greater difficulty for ORCs? Why is the predicted difference at *that* exactly zero?

```
# Your code here:
# expectations$src_bits <- ...
# expectations$orc_bits <- ...
# expectations$diff_bits <- ...
print(expectations[, c("region", "src_prob", "orc_prob")])
```

```
##       region     src_prob     orc_prob
## 1       that 0.0021913142 0.0021913142
## 2 determiner 0.1131102965 0.0230793237
## 3       noun 0.0020720549 0.0016407099
## 4    RC verb 0.0016569279 0.0006965855
## 5  main verb 0.0003000433 0.0001159790
```

Question 7 Compare the **direction and relative size** of `diff_bits` with the observed `diff_gopast` from Part 2. Plot the two variables against one another (one point per region), and label the points. Do larger differences in GPT-2 surprisal generally correspond to larger observed ORC reading-time costs? Give

two reasons why this exploratory plot does not amount to a convincing independent test of the surprisal theory.

```
comparison <- merge(reading, expectations, by = "region", sort = FALSE)
# Your code here; hints:
# plot(comparison$diff_bits, comparison$diff_gopast,
#      xlab = "ORC - SRC GPT-2 surprisal (bits)",
#      ylab = "ORC - SRC go-past time (ms)", pch = 16)
# text(comparison$diff_bits, comparison$diff_gopast,
#      labels = comparison$region, pos = 3, cex = 0.8)
```

Part 4: Model interpretation (5 minutes)

Question 8 Suppose the GPT-2 predictions capture increased surprise at an ORC determiner, but do not fully explain the extra reading time at the RC verb. Suggest **one additional cognitive mechanism**, beyond expectation/surprisal, that might explain this mismatch. How would you design a stronger empirical test to distinguish these explanations?

If you have extra time: Recompute the surprisal differences using `-log()` (natural logarithms) instead of `-log2()`. What changes, and what stays the same?

References

Radford, Alec, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. "Language Models Are Unsupervised Multitask Learners." OpenAI. https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

Staub, Adrian. 2010. "Eye Movements and Processing Difficulty in Object Relative Clauses." *Cognition* 116 (1): 71–86. https://doi.org/10.1016/j.cognition.2010.04.002.

Staub, Adrian, and Charles Clifton Jr. 2006. "Syntactic Prediction in Language Comprehension: Evidence from Either ... Or." *Journal of Experimental Psychology: Learning, Memory, and Cognition* 32 (2): 425–36. https://doi.org/10.1037/0278-7393.32.2.425.