Computational Cognitive Science 2026/27

Week 9

Frank Keller

Solutions for 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 must be placed in the same working directory as this Rmd 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) {
  -log2(p)
}
```

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

```
##     word probability     bits
## 1 butter       0.500 1.000000
## 2    jam       0.250 2.000000
## 3 cheese       0.125 3.000000
## 4  socks       0.001 9.965784
```

Solution: *The most surprising word is* socks *(about 9.97 bits); the other values are 1, 2, and 3 bits, respectively. A halving of probability adds exactly one bit of surprisal. According to the surprisal hypothesis,* socks *should be the most difficult continuation, all else equal.*

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
s_without_either <- surprisal(p_without_either)
s_with_either <- surprisal(p_with_either)
print(c(without_either = s_without_either, with_either = s_with_either))
```

```
## without_either    with_either
##       3.321928       1.321928
```

```r
print(s_without_either - s_with_either)
```

```
## [1] 2
```

Solution: *Surprisal decreases from about 3.32 to 1.32 bits (a decrease of 2 bits). We therefore predict shorter reading times when* either *is present, consistent with a prediction-based explanation of the result discussed by (Staub and Clifton 2006). These probability values were supplied for illustration, not estimated from that study.*

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")
```

```r
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?

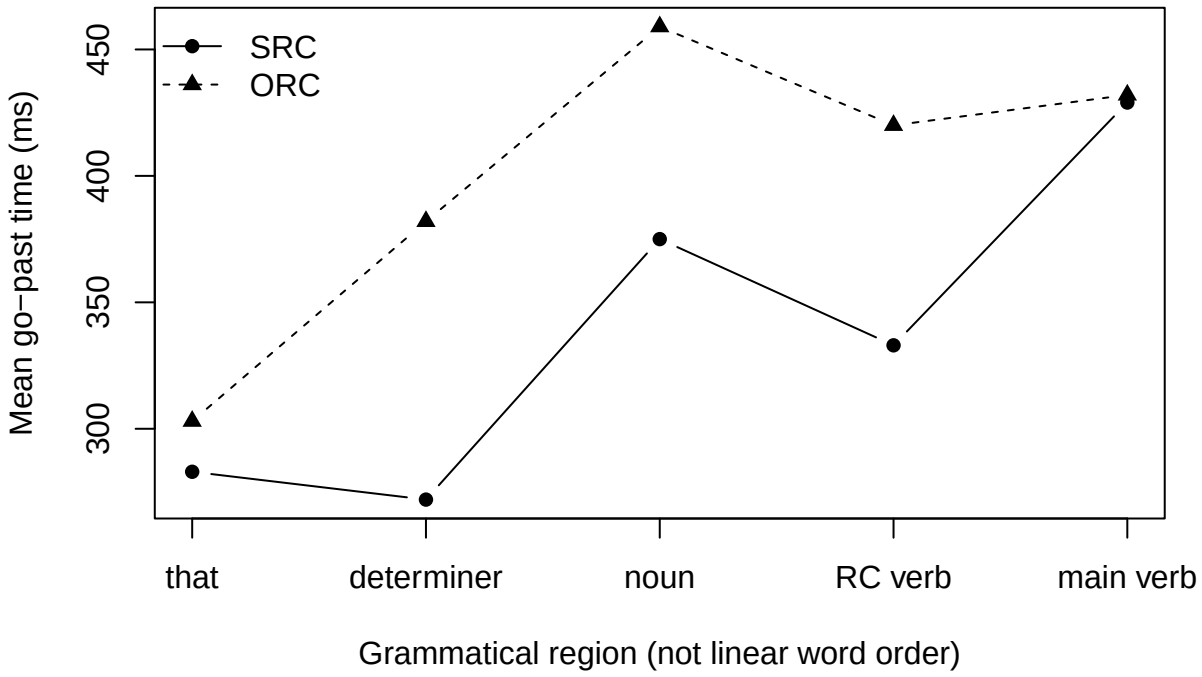
```r
reading$diff_gaze <- reading$orc_gaze - reading$src_gaze
reading$diff_gopast <- reading$orc_gopast - reading$src_gopast
print(reading[, c("region", "diff_gaze", "diff_gopast")], row.names = FALSE)
```

```
##      region diff_gaze diff_gopast
##        that        13          20
##  determiner        10         110
##        noun       -50          84
##     RC verb        48          87
##   main verb       -11           3
```

Solution: *For gaze duration, the differences (ORC minus SRC) are 13, 10, -50, 48, and -11 ms, respectively. The largest ORC cost is at the RC verb (48 ms); at the noun, ORCs actually have shorter gaze durations. For go-past time, the differences are 20, 110, 84, 87, and 3 ms. The largest cost is now at the determiner (110 ms), followed by the RC verb (87 ms) and noun (84 ms). Thus, the pattern depends considerably on the measure.*

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.

```r
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$region)
lines(x, reading$orc_gopast, type = "b", pch = 17, lty = 2)
legend("topleft", c("SRC", "ORC"), pch = c(16, 17),
       lty = c(1, 2), bty = "n")
```



Solution: *The ORC line is above the SRC line throughout, but the difference is much larger at the determiner, noun, and RC verb than at* ***that*** *or the main verb. The x-axis aligns grammatical regions across conditions; it does not show their actual order of presentation.*

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?

Solution: *At the noun, ORC gaze duration is* ***50 ms shorter*** *than SRC gaze duration, but ORC go-past time is* ***84 ms longer****. At the RC verb, ORC gaze duration is* ***48 ms longer****, while ORC go-past time is* ***87 ms longer****. The larger go-past costs suggest that regressions or re-reading contribute to difficulty, particularly around the noun; however, these aggregate measures alone cannot identify exactly when or where regressions occurred. The five rows are five* ***regions****, not five independently sampled people. They are condition means aggregated across trials and participants, so treating them as independent observations would misrepresent the sampling variability and produce invalid significance tests.*

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.

```r
# 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?

```r
expectations$src_bits <- surprisal(expectations$src_prob)
expectations$orc_bits <- surprisal(expectations$orc_prob)
expectations$diff_bits <- expectations$orc_bits - expectations$src_bits
print(expectations[, c("region", "word", "src_bits", "orc_bits", "diff_bits")],
      row.names = FALSE)
```

```
##      region     word  src_bits  orc_bits diff_bits
##        that     that  8.833988  8.833988  0.000000
##  determiner      the  3.144198  5.437255  2.293057
##        noun  senator  8.914722  9.251464  0.336742
##     RC verb attacked  9.237273 10.487412  1.250138
##   main verb admitted 11.702542 13.073849  1.371307
```

```r
print(expectations$region[expectations$diff_bits > 0])
```

```
## [1] "determiner" "noun"       "RC verb"    "main verb"
```

Solution: *GPT-2 predicts higher surprisal for ORCs at the* ***determiner***, ***noun***, ***RC verb***, *and* ***main verb***. *The largest difference is at the determiner (about* ***2.29 bits***), *followed by the main verb (****1.37 bits***), *RC verb (****1.25 bits***), *and noun (****0.34 bits***). *At* ***that***, *the difference is exactly zero because both conditions have the identical preceding context (****The reporter***) *and the same target word.*

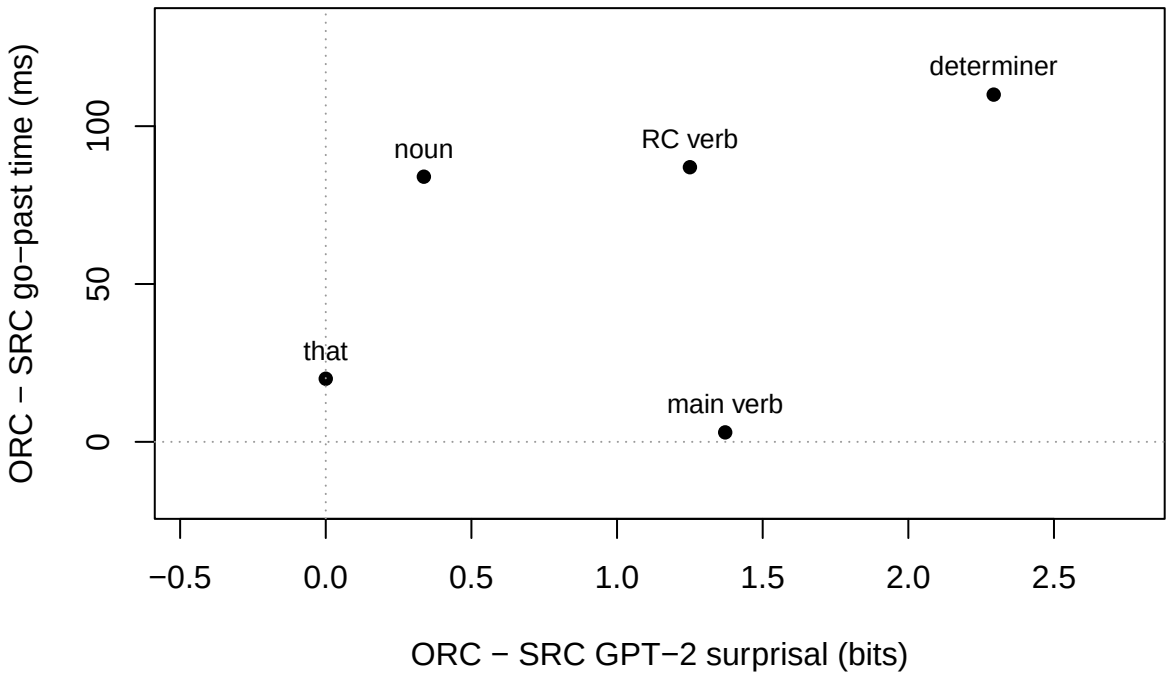
Region	SRC surprisal (bits)	ORC surprisal (bits)	ORC - SRC (bits)
that	8.83	8.83	0.00
determiner	3.14	5.44	2.29
noun	8.92	9.25	0.34
RC verb	9.24	10.49	1.25
main verb	11.70	13.07	1.37

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)
# Restore the grammatical-region order after merging.
comparison <- comparison[match(regions, comparison$region), ]
print(comparison[, c("region", "diff_bits", "diff_gopast")],
      row.names = FALSE)
```

```
##      region diff_bits diff_gopast
##        that  0.000000          20
##  determiner  2.293057         110
##        noun  0.336742          84
##     RC verb  1.250138          87
##   main verb  1.371307           3
```

```
plot(comparison$diff_bits, comparison$diff_gopast,
     xlab = "ORC - SRC GPT-2 surprisal (bits)",
     ylab = "ORC - SRC go-past time (ms)", pch = 16,
     xlim = extendrange(comparison$diff_bits, f = 0.2),
     ylim = extendrange(comparison$diff_gopast, f = 0.2))
text(comparison$diff_bits, comparison$diff_gopast,
     labels = comparison$region, pos = 3, cex = 0.8)
abline(v = 0, h = 0, lty = 3, col = "gray60")
```



Solution: *There is **partial, but not consistent, agreement**. GPT-2 predicts its largest ORC surprisal cost at the determiner (**2.29 bits**), which also has the largest observed go-past cost (**110 ms**). It predicts an ORC cost at the RC verb (**1.25 bits**), where humans show an **87 ms** cost. However, GPT-2 also predicts a substantial surprisal cost at the **main verb** (**1.37 bits**) despite an observed difference of only **3 ms**. Conversely, the noun has a substantial human cost (**84 ms**) but only a small predicted surprisal difference (**0.34 bits**). Thus, a simple 'more surprisal = proportionally more reading time' account does not fit all five*

regions well.

Two important limitations are (1) ***stimulus mismatch****: GPT-2 scores a single illustrative sentence pair whereas the human data average over different experimental sentences, and (2)* ***aggregation/non-independence****: the five points are grammatical regions, not independent participant-level observations. Other influences on reading time, including reanalysis and regressions, are also unaccounted for.*

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?

Solution: *One possibility is* ***memory-based retrieval/integration difficulty****: at the RC verb, the reader must retrieve the displaced object of the verb and integrate it with the clause structure, potentially increasing processing time beyond what next-word surprisal predicts. Alternatively, costly syntactic reanalysis or eye-movement regressions may contribute. A stronger test would obtain GPT-2 (and ideally other language-model) probabilities for* ***all the actual experimental items****, match predicted surprisal to the relevant word/region for each item, and test reading times with participant- and item-level data. A regression or mixed-effects model could compare surprisal alone with surprisal plus a principled memory-distance/retrieval predictor, while controlling for factors such as word length and frequency. Evidence that the additional predictor explains reliable residual variation would help distinguish the accounts.*

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

Solution: *The unit changes from bits to natural units (nats): `-log(p) = log(2) * (-log2(p))`. Every surprisal and surprisal difference is multiplied by the same positive constant, `log(2)` (approximately 0.693). Therefore rankings, signs, and the shape of the comparison remain unchanged, apart from a rescaling of the horizontal axis.*

```r
# Optional check: changing from bits to nats preserves the signs.
if (exists("expectations")) {
  diff_nats <- -log(expectations$orc_prob) + log(expectations$src_prob)
  stopifnot(isTRUE(all.equal(diff_nats, log(2) * expectations$diff_bits)))
}
```

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.