Informatics 1 Cognitive Science – Tutorial 4 Solutions

Frank Keller, Carina Silberer, Frank Mollica

Week 5

1 Word Learning

In the lectures, we discussed Piantadosi et al.'s program induction model for learning number words. The model considers several possible hypotheses for the definition of number words and uses Bayes Rule to infer the most likely hypothesis as a learner sees data. Recall Bayes rule:

$$P(h|d) = \frac{P(d|h)P(h)}{\sum_i P(d|h_i)P(h_i)}. \tag{1}$$

Exercise 1 and solution We will attempt to replicate Piantadosi et al.'s results. Instead of searching a vast hypothesis space at multiple data amounts, let's focus on the four hypotheses in Figure 1 and the two datasets in Table 1.

One-knower

```
λ S . (if (singleton? S)
          "one"
          undef)
```

Two-knower

```
λ S . (if (singleton? S)
          "one"
          (if (doubleton? S)
              "two"
              undef))
```

Three-knower

```
λ S . (if (singleton? S)
          "one"
          (if (doubleton? S)
              "two"
              (if (tripleton? S)
                  "three"
                  undef)))
```

CP-knower

```
λ S . (if (singleton? S)
          "one"
          (next (L (set-difference S
                                   (select S)))))
```

Figure 1: The four hypotheses for number word meanings that we are considering.

Dataset	(one, ·)	(two, :)	(three, :·)	(four, ::)	(five, ::·)	(six, :::)	(seven, :::·)	(eight, ::::)
($N = 30$)	20	4	2	2	0	1	0	1
($N = 60$)	41	10	5	2	0	1	0	1

Table 1: The datasets we are considering. Each column denotes a type of data. Each row contains the number of times that type of data has been seen.

1. In the original model, they used a simplicity prior. Let's try using a prior based on the description length of the hypotheses. For each hypothesis, write down the description length (include parentheses and punctuation).

   One-knower: 30 characters;
   Two-knower: 52 chars;
   Three-knower: 75 chars;
   Cardinal Principle: 60 chars.

2. Let's say the prior probability of a hypothesis is inversely proportional to its description length. So longer description lengths are less probable a priori.

$$P(h) = \frac{{DL_h}^{-1}}{\sum_i {DL_{h_i}}^{-1}} \tag{2}$$

   Fill in the prior in Tables 2 and 3.

3. In the original model, they used a noisy size principle likelihood, which considers two possible ways the data might have been generated: i) according to the hypothesis and ii) randomly correct. Formally the noisy size principle is:

$$P(w|h,s) = \begin{cases} \alpha + \frac{1-\alpha}{10} & \text{if } w = h(s) \\ \frac{1-\alpha}{10} & \text{else} \end{cases} \tag{3}$$

   The parameter α reflects how reliably the data comes from the hypothesis. Let's consider $\alpha = 0.9$ for this exercise. Fill in the rest of the likelihoods in Table 3.

4. Now use Bayes Rule to calculate the posterior beliefs over knower levels. In the real world, we often deal with probabilities of events that are really small. To make the calculations easier we can work in logarithms. Here is the log of Bayes rule:

$$\log P(h|d) = \log P(d|h) + \log P(h) - \log P(d), \tag{4}$$

   where $P(d) = \sum_h \exp(\log P(d|h) + \log P(h))$.

   Most programming languages have a function that computes this `LogSumExp` operation. To make your life easier, the attached python script walks through this computation. You should be able to plug in the above information and it will return the posterior. You can run it locally or copy it on notable.

5. Take a look at the results from the original model (in the slides). Did we replicate their results? Was description length an appropriate prior? Take a guess on how we might have to change the prior.

   Solution We did not replicate. While description length is consistent with simplicity, this implementation does not replicate the original model. What might be done to fix this? Well we need to penalize the CP-knower hypothesis. This is exactly what the original model did, noting that recursive reasoning is a particularly difficult operation. For fun you could implement this bias and add -45 to the CP-knower hypothesis prior.

Knower Level	Prior	Likelihood$_{N=30}$	Posterior
1-knower	0.4037	$(0.91)^{20}(0.01)^{10}$	0.0000
2-knower	0.2329	$(0.91)^{24}(0.01)^{6}$	0.0000
3-knower	0.1615	$(0.91)^{26}(0.01)^{4}$	0.0000
CP-knower	0.2019	$(0.91)^{30}(0.01)^{0}$	0.9999

Table 2: Use this table to write down the components of Bayes Rule for the first dataset $N = 30$.

Knower Level	Prior	Likelihood$_{N=60}$	Posterior
1-knower	0.4037	$(0.91)^{41}(0.01)^{19}$	0.0000
2-knower	0.2329	$(0.91)^{51}(0.01)^{9}$	0.0000
3-knower	0.1615	$(0.91)^{56}(0.01)^{4}$	0.0000
CP-knower	0.2019	$(0.91)^{60}(0.01)^{0}$	0.9999

Table 3: Use this table to write down the components of Bayes Rule for the second dataset $N = 60$.

2 Communication and Efficiency

In class we discussed the mathematical model of communication and how it has been used in computational psycholinguistics to predict processing difficulties measured via reading times. The aim of this exercise is to explore how we can use information theory to make predictions about processing difficulties.

Consider the following sentences:

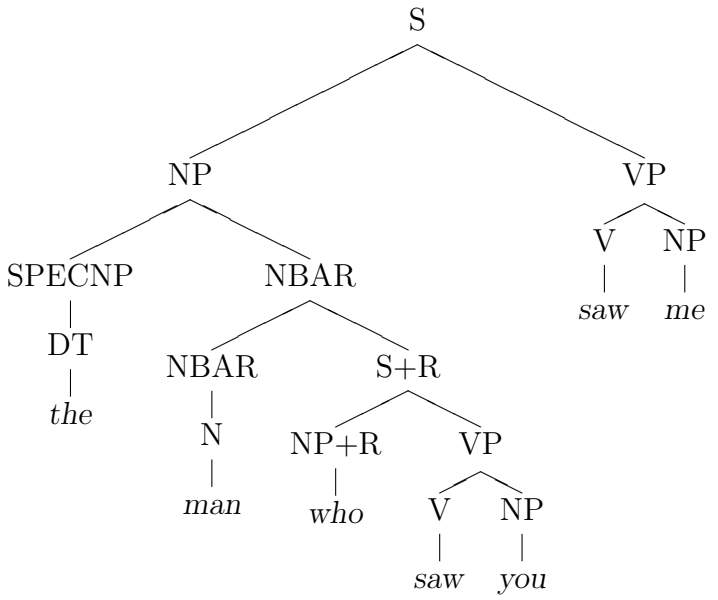
(1) The man who saw you saw me.

(2) The man who you saw saw me.

The first sentence is called a subject relative clause because *the man* is the subject of the relative clause. The second sentence is called an object relative clause because *the man* is the object of the relative clause.

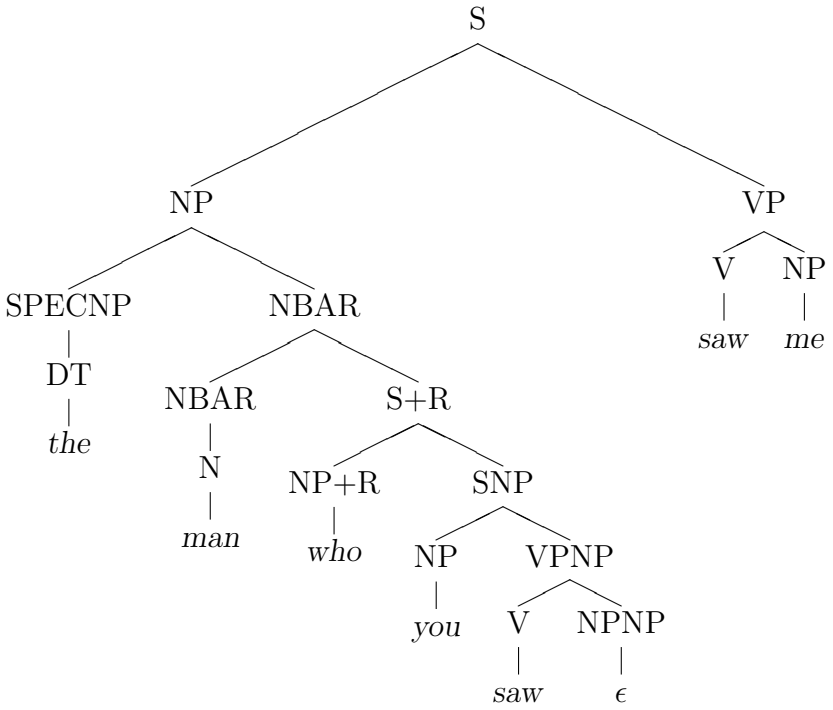
Exercise 2

1. Figure 2 contains a grammar sufficient to parse both trees. ϵ means an empty space. Draw the tree structures for both sentence.



NP	→	SPECNP NBAR
NP	→	*you*
NP	→	*me*
SPECNP	→	DT
NBAR	→	NBAR S[+R]
NBAR	→	N
S	→	NP VP
S[+R]	→	NP[+R] VP
S[+R]	→	NP[+R] S/NP
S/NP	→	NP VP/NP
VP/NP	→	V NP/NP
VP	→	V NP
V	→	*saw*
NP[+R]	→	*who*
DT	→	*the*
N	→	*man*
NP/NP	→	ϵ

Figure 2: A grammar learned from a large corpus of hand-annotated parses.



2. Psycholinguists have found that one type of relative clause is more difficult to process than the other. Based on the tress, take a guess as to which one is more difficult. Justify your response.

   Object relative could be harder because need to introduce a null word ϵ to parse or because it uses more rules.

3. In class, we saw that reading time was a measure of processing difficulty and that reading time is proportional to the surprisal of a word in context (i.e., $-\log_2 P(w|c)$). That means that words with higher surprisal (information content) take longer to read. Table 4 contains the trigram predictability for the words in the example sentences. Trigram probabilities $P(w_3|w_1, w_2)$ reflect the probability of a word w_3 given the two words that precede it w_1 and w_2 as context. Plot out the bar chart of each sentences' word by word surprisal in Figure 3.

4. Does this change your opinion of which type of relative clause is more difficult to process? Why?

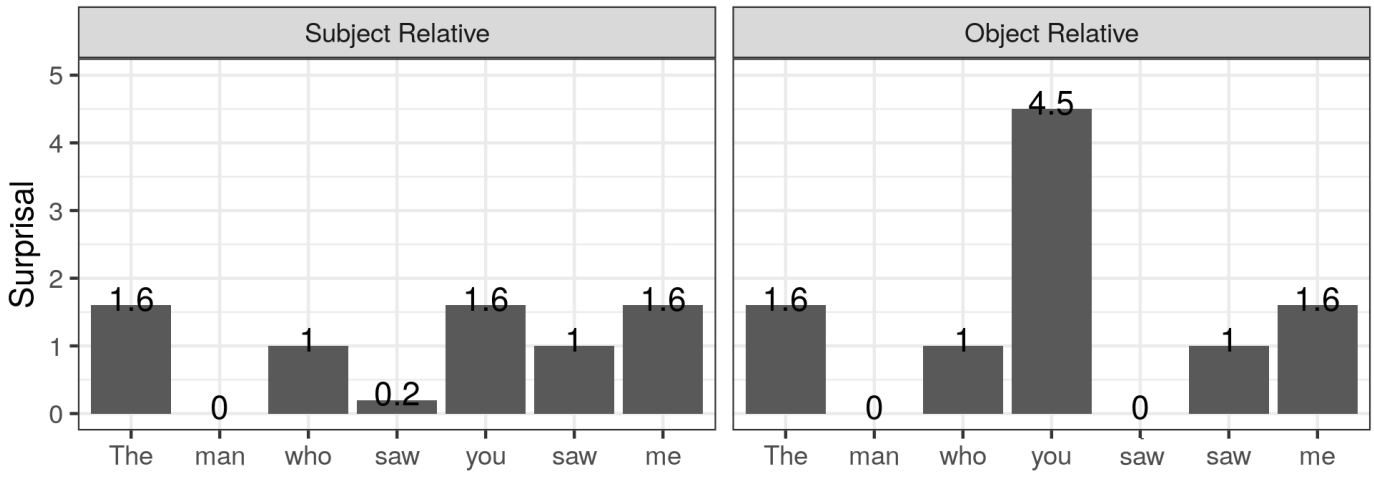


Figure 3: Solution plot of the surprisal.

$P(the\mid \#, \#)= 0.329877$	$P(man\mid \#, the) = 1$	$P(who\mid the, man) = 0.5$
$P(you\mid who, saw) = 0.329877$	$P(saw\mid saw, you) = 0.5$	$P(me\mid you, saw) = 0.329877$
$P(saw\mid who, you) = 1$	$P(saw\mid you, saw) = 0.5$	$P(me\mid saw, saw) = 0.329877$
$P(saw\mid man, who) = 0.8687422$	$P(you\mid man, who) = 0.04419417$	

Table 4: Trigram probability for each word in the sentence.

This is more evidence that the object relative is harder to process because *you* is surprising and therefore should incur processing difficulties and longer reading times.