Appendix A

Probabilities, parameters, models and grammars

In this appendix we ask linguists to put themselves in the learner's shoes, and to run through some calculations as if they go through the experience themselves in a deliberately mechanistic and simplified world.

Consider a hypothetical joint probability distribution of logical forms and sentences. We can write it as $P(L,S)$, where L is a random variable over all logical forms in all languages, and S is a random variable over all sentences.

Finding such a joint distribution which is consistent with language use would be like living in a Chomskyan paradise where arbitrary sentences somehow line up with right meanings which are always faithful to the alignment in the language being exposed to. We can think of this scenario as $P(L,S) = P(L)$, that is, $P(S \mid L) = 1$. It does not seem very likely that the paradise will include linguist's understanding of the child's understanding of the Gavagai problem before his second birthday under exposure to some native language, which seems to be the timeframe for language acquisition across cultures and different upbringings as reported since Lenneberg (1967).

The other extreme, which we may call the Quinean paradise, would be where $P(L,S) = P(S)$, that is, $P(L \mid S) = 1$. This is where the child is on the mark for the expression L for every exposure to the random variable S. The linguist's paradise achieved becomes the child's nightmare world, because now he is expected to know that there isn't anything confusing about the states of affairs described by *Do you know that it is bedtime?* and *Do you know that dog?*, or that *Gavagai* might mean in the adult world black spots on the visible hind legs of a rabbit passing by.

The asymmetry in these extremes is that it did not seem too difficult to convince Quine that a logician's paradise does not exist, in fact he himself said so, that it does not seem to be a logical possibility,[1] rather than considering it a syntactic improbability as we do, it has proven very difficult to convince Chomsky that his paradise does not exist either.

The real world that children face seems to lie somewhere in between. The Cartesian deterministic world waiting to be discovered by perfect creatures, that is, by creatures with perfect interfaces according to Chomsky (2001), because of language, doesn't exist. (Descartes was more cautious than his followers; he suggested in 1701/1954:177–78 to divide knowledge into simple

1. "[L]et us recall the predicament in radical translation, which showed that a full knowledge of the stimulus meaning of an observation sentence is not sufficient for translating or even spotting a term." Quine 1960:236.

propositions and problems, and the latter to ones perfectly understandable and which aren't—the latter work was never finished but clearly intended, to be tackled by different sets of "rules" by his own admission.) The perfect logic of nature out there waiting to be discovered by imperfect creatures, in the terminology of Carnap 1937/1967:2, does not exist either, nor chaos.

Working toward a more first-order realistic world we must realize that the child is facing a conditional probability problem over what is said and what is meant by what is said, rather than a joint probability problem, over the symbolic species that we call grammar. In simplified terms, the problem is fitting a good $P(L \mid S)$ for understanding by the child, and $P(S \mid L)$ for production, given some $P(L \mid S)$.

One question then is the following: What is the mechanism that does the conditioning? We suggest it is latent syntax. It is easier to tackle this question for $P(L|S)$ because recognition has been better understood than generation. The discussion might dispell one common misconception in linguistics about the role of syntax in probabilistic inference, that it would seem not to rely on theory. It will also exemplify realizations of every single degree of freedom we have made use of in this book, in a computational model.

A.1 Hypotheses in the grammar

Consider the two entries below. They can be further distinguished, say as assertion of proposition s in the second case. We keep them simple.

(266) a. knows := $(S\backslash NP)/NP$: $\lambda x \lambda y.know' xy$
b. knows := $(S\backslash NP)/S$: $\lambda s_e \lambda y.know' s_e y$

We can start distancing ourselves from both paradises by assuming that nothing stops misunderstanding the two uses of *know* from entering grammar. For this controlled experiment let us assume they are:

(267) c. knows := $(S\backslash NP)/NP$: $\lambda x \lambda y.know' yx$?
d. knows := $(S\backslash NP)/S$: $\lambda s_e \lambda y.know' y s_e$?

Similary, let us assume *love* can present a similar puzzle:

(268) e. loves := $(S\backslash NP)/NP$: $\lambda x \lambda y.love' yx$
f. loves := $(S\backslash NP)/NP$: $\lambda x \lambda y.love' yx$?

As for participants of these states of affairs, let's also assume that our capture of them in the grammar is half the times wrong. The entries below with '?' are

not proper type-raising according to the current theory. But let's assume they are all in the grammar.

(269) g. John := $S/(S\backslash NP) : \lambda p.pj'$
h. John := $S/(S\backslash NP) : j'$?
i. John := $(S\backslash NP)\backslash((S\backslash NP)/NP) : \lambda p.pj'$
j. John := $(S\backslash NP)\backslash((S\backslash NP)/NP) : j'$?
k. Mary := $S/(S\backslash NP) : \lambda p.pm'$
l. Mary := $S/(S\backslash NP) : m'$?
m. Mary := $(S\backslash NP)\backslash((S\backslash NP)/NP) : \lambda p.pm'$
n. Mary := $(S\backslash NP)\backslash((S\backslash NP)/NP) : m'$?

Finally, let us introduce a Quinean complication. The learner might think that *John* means dog' and *Mary* means cat', perhaps because there were such kinds of animals around when these words were uttered:

(270) o. John := $S/(S\backslash NP) : \lambda p.p\,dog'$?
p. John := $(S\backslash NP)\backslash((S\backslash NP)/NP) : \lambda p.p\,dog'$?
q. Mary := $S/(S\backslash NP) : \lambda p.p\,cat'$?
r. Mary := $(S\backslash NP)\backslash((S\backslash NP)/NP) : \lambda p.p\,cat'$?

Both paradises must live with these categories because they cannot be purged by them. They are empirical "mistakes."

Let the grammar consist of only the entries above, from (266).(a) to (r) above. Since it manifests a great deal of uncertainty (11 items out of 18 are not suitable for adult English which communicated the states of affairs to the learner), and in the beginning we don't really know why the examples which are not marked '?' are better than the others, we must have some weights associated with these entries. Let them all be equal, say unity, to avoid any preexposed bias toward plain old English. These are the initial values of the weights, and a uniform prior seems reasonable.

A.2 Exposed data

If the bearer of grammar (a-r) is being exposed to English, he would be hearing examples like below (avoiding other complications and sources of uncertainty to make our point as quickly as possible):[2]

2. This is another way to say that we don't assume they already know English to learn better English. The grammar itself contains *hypotheses* on behalf of the bearer as to what parts of strings

(271) 1. John loves Mary : $love'm'j'$
2. Mary loves John : $love'j'm'$
3. John knows Mary : $know'm'j'$
4. Mary knows John : $know'j'm'$
5. Mary knows John loves Mary : $know'\ (love'm'j')\ m'$
6. John knows Mary loves John : $know'\ (love'j'm')\ j'$

Notice that there are no syntactic assumptions about what is assumed to be heard and inferred. Expressions themselves are paired with logical forms. We are not even told which fragments in the expressions gave the LF objects, say $know'$.

This is where syntax comes in. There are many variations on the theme of bringing syntax in to conditional probabilities. We will exemplify from the simplest mechanism of Probabilistic CCG (Clark and Curran, 2003; Zettlemoyer and Collins, 2005), where parameters are associated with lexical items just like weights above (more on parameters as species below). For brevity we will eschew how they generate the lexicon in the first place, figuring out word boundaries, etc.; see the last footnote.

A.3 The model

The expressions above to the left of ':' are evaluated for the logical forms they deliver. Because the lexical items have multiple categories, we must measure their contribution to (271). We are looking for the most likely LF given a sentence, based on the fact that we have been through (271) with the hypotheses in (a–r). This LF is the one with the highest conditional probability on the left side of the equation below (there are alternative methods):

(272) $\arg\max_L P(L \mid S;\bar{\theta}) = \arg\max_L \sum_D P(L,D \mid S;\bar{\theta})$

Because the logical form is obtained only through derivations D, because of (1), i.e. by syntax, and because of the fact that the same L can be revealed by more than one derivation, we actually use the equation on the righthand side, which gives the LF that has the best total probability in all derivations of L. $\bar{\theta}$ here is the parameter vector in the sense the term is understood in probability theory and statistics. In this particular example it has the size of 18,

could mean what. As we suggested in chapter 7, they probably start with a small and tight correspondence of semantic types and syntactic types to bootstrap the process. Abend et al. 2016 show one particular model for that.

from (a) to (r) in the grammar. Initially its value is eighteen 1s because of our assumptions.

The weights in $\bar{\theta}$ are not probabilities. One simple way to turn them into probabilities, making explicit use of syntax, is counting the number of times a lexical item is used in a derivation D, divided by total sum for all derivations. The rationale for this method is based on the fact that if a lexical hypothesis is the right one to lead to a highly probable LF, then its use must be causing it. Other measures are of course possible as long as they give a probability sum of 1 for all derivations for S. Below $\bar{f}$ is a vector (size 18) of 3-argument functions $<f_1(L,D,S),\cdots f_{18}(L,D,S)>$ that gives such counts for every triple L,D,S:

$$(273)\quad P(L,D \mid S;\bar{\theta}) = \frac{e^{\bar{f}(L,D,S)\cdot\bar{\theta}}}{\sum_L \sum_D e^{\bar{f}(L,D,S)\cdot\bar{\theta}}}$$

The crucial effect of (271) on (a–r) is the changes in parameters that are implicated by it. One way to measure that is to look at the local gradient of (271) with respect to the parameter θ_j for lexical item j. This is a local difference in two expected values where the overall training set is $O_i = (L_i, S_i)$ in (271) (again, there are alternatives):

$$(274)\quad \frac{\partial O_i}{\partial \theta_j} = E_{f_j(L_i,T,S_i)} - E_{f_j(L,T,S_i)}$$

If lexical item j participates in more correct derivations than incorrect ones, local derivative of θ_j will be positive. If all derivations with it are correct then there will be no change. If it takes part more in incorrect derivations it will be negative. The expected values E of f_j are therefore calculated under the distributions $P(T \mid S_i, L_i; \bar{\theta})$ for the first term in (274), and $P(L,T \mid S_i; \bar{\theta})$ for the second. We can update the parameters based on these gradients. One method is given in Zettlemoyer and Collins (2005).

For (a–r) above starting with unit parameters, and subjected to (271) for update, we get the results in Table A.1 (in this experiment 10 passes were done over (271), and all learning parameters mentioned by Zettlemoyer and Collins were set to 1.0).

We can see that ill-fated subject type-raising of John and Mary, entries (h) and (l), are penalized by (271). Likewise ill-fated object-type-raising of John and Mary, entries (j) and (n). The second entry for *loves*, (f), is disfavoured too,

Table A.1: Training the grammar (a-r) on the data in (271).

index	item	initial value	final value	difference
a	knows	1.0	5.29	4.29
b	knows	1.0	4.62	3.62
c	knows	1.0	-3.29	-4.29
d	knows	1.0	-2.62	-3.62
e	loves	1.0	7.48	6.48
f	loves	1.0	-5.48	-6.48
g	John	1.0	6.58	5.58
h	John	1.0	-1.79	-2.79
i	John	1.0	8.53	7.53
j	John	1.0	-2.76	-3.76
k	Mary	1.0	6.4	5.4
l	Mary	1.0	-3.32	-4.32
m	Mary	1.0	9.64	8.64
n	Mary	1.0	-3.328	-4.32
o	John	1.0	-1.8	-2.8
p	John	1.0	-2.76	-3.76
q	Mary	1.0	-1.7	-2.7
r	Mary	1.0	-3.32	-4.32

because of its very un-English LF. In one sense, the two entries for *loves* aren't really competing with each other because (f) never gives the right LF in (271) but (e) does. Similarly, the entries (a) and (b) for *knows* are not competing, but (c) and (d) do compete with them. Entries (a) and (b) provide right LFs in (271) but for different subcategorizations of *knows*.

The Quinean "errors" in the grammar, (o–r), which are well-formed and satisfy the constraints of the theory but make wrong assumptions about semantics, are penalized by (271) too, as shown in the bottom four rows of the table.

In this deliberately simplifed world where the hypotheses in (a–r) are given 6 opportunities in (271) 10 times to stand against data (this is because the gradients are local to parameters, so we need to *estimate* the overall gradient of the model), we can already see the difference. Abend et al. (2016) show that the process carried over longer sequences of acquisition can give the impression of making a switch-like jump to the adult correspondence rather than simple ranking by the derivational process every time, which is what really takes place under the hood.

The thought experiment has been implemented computationally. It is reported at `github.com/bozsahin/ccglab-models`, in the model `noqnoc`.