Sentence Processing
Part 3

Emily Morgan
LSA 2019 Summer Institute
UC Davis
So far

What makes words and sentences easier/more difficult to process?

Evidence for both the memory-limitation and expectation-based theories
Quantifying online processing difficulty

• So far, we’ve evaluated *qualitative* predictions of the memory-limitation and expectation-based theories (i.e. predictions of faster vs. slower)

• But we can use our probabilistic models to make *quantitative* (i.e. numeric) predictions, particularly for the expectation-based theory
  • The language models we discussed before (e.g. n-grams, PCFGs) give us P(word|context)
Surprisal($w_i$) \equiv \log \frac{1}{P(w_i|\text{CONTEXT})} = -\log P(w_i|\text{CONTEXT})

\approx -\log P(w_i|w_1...i-1)

- Intuitively, a measure of how much surprise a word elicits
- Claim: Reading time($w_i$) \propto \text{Surprisal($w_i$)}
- Captures the expectation-based theory intuition: the more expected a word (i.e. the less surprising), the less processing difficulty

(Hale, 2001; Levy, 2008)
Why surprisal?

• Formally, surprisal is a measure of the information content of a word
  • Who is also taking Richard Futrell’s Information Theoretic Approaches to Language class?
• Let’s unpack that...
A very simple generative model of sentence processing

Speaker intends a tree $T$

Comprehender hears a sentence $S$ (equivalently, words $w_1$ through $w_n$)

Comprehender’s goal: Infer $T$ from $S$, i.e. calculate the distribution $P(T|S)$
The dog near the rat and the cat slept

Deterministic

The dog near the rat and the cat slept
The dog near the rat and the cat slept

Hence, comprehender must infer the full probability distribution $P(T|S)$ for all possible trees $T$.
Incremental comprehension

- To make it still more complicated, remember that sentence processing is incremental.
- So comprehenders must infer the probability over trees $T$ on the basis of incomplete input, i.e. $P(T|w_{1...i})$ where $w_{1...i}$ is not the full sentence.
- e.g. Suppose you have only seen three words The dog near, what are the probabilities of all possible trees that yield sentences that begin with those words?
  - i.e. Must infer $P(T|The\ dog\ near)$ for all possible trees $T$.
- We notate this distribution as $P_i^T \equiv P(T|w_{1...i})$. 
Surprisal theory’s main claim

• The difficulty of processing a word is quantified by how much $P^T$ must be updated, e.g. the difference between $P^T_{i-1}$ and $P^T_i$

• How do we quantify the difference between two probability distributions?
Relative entropy

The *relative entropy* of a distribution $q$ with respect to a distribution $p$, aka. the *Kullback-Leibler (KL) divergence* of $q$ from $p$, is:

$$D(q||p) = \sum_{T \in \mathcal{T}} q(T) \log \frac{q(T)}{p(T)}$$

- Intuitively, this is a measure of how hard it is to encode the distribution $q$ using the distribution $p$
  - e.g. if a particularly tree $T'$ is high probability in $p$, it’s easy to encode, and if it’s low probability in $p$, it’s hard to encode
  - So $q$ is easier to encode if high probability trees in $q$ are also high probability in $p$
- If $q=p$, $D(q||p) = 0$
- As $q$ and $p$ diverge more, $D(q||p) \to \infty$
Relative entropy $\rightarrow$ Surprisal

- So difficulty of processing word $i$ is predicted to be $D(P_i^T \mid\mid P_{i-1}^T)$
- Mathematically, it turns out that $D(P_i^T \mid\mid P_{i-1}^T) = \text{Surprisal}(w_i) = -\log P(w_i \mid w_1 \ldots w_{i-1})$
  - (See Levy, 2008 for proof)
- This is true regardless of the exact structure of the underlying form $T$ or the grammar that generates it
  - Could be n-grams, PCFGs, RNNs, etc.
- So difficulty of processing word $i$ is equivalently predicted to be $\text{Surprisal}(w_i) = -\log P(w_i \mid w_1 \ldots w_{i-1})$
Technical details

• We typically calculate surprisal using $\log_2$:
  \[
  \text{Surprisal}(w_i) = -\log_2 P(w_i | w_{1..i-1})
  \]

• In this case, the unit for surprisal is **bits**
  • aka. binary digits or 1/0, like a computer

(For more on the relationship between probability and information content, see Goldsmith, 2007, article or MacKay, 2003, textbook)
Review: Claims of surprisal theory

- The goal of sentence processing is to infer the probability distribution over possible trees $P_T$ given the input (incrementally at each word).
- Difficulty of processing a word is quantified by how much $P_T$ must be updated:
  - i.e. the “work” of sentence processing is updating distributions over incremental parses.
  - quantified via relative entropy, aka. KL-divergence, $D(P_T^i || P_T^{i-1})$.
- Mathematically, how much $P_T$ must be updated upon receiving word $i$ is equal to the surprisal of word $i$ in context, $-\log P(w_i | w_1...i-1)$.
Implications of surprisal theory (if true)

• Effectively, surprisal is a measure of the marginal probability of the next word in a sentence (marginalizing over all possible trees)

• So “predicting” the next word in a sentence is implicit in calculating probabilities over possible tree structures
  • A potential answer to some people questions about why the parser would “waste resources” on predicting upcoming words

• Conversely, the difficulty of processing a possible tree structure (including its syntactic structure) “bottoms out” in the probability of processing each word incrementally
  • Calculating the probability of each incoming word doesn’t ignore syntactic structure; it implicitly takes into account how likely different syntactic structures are to generate that word
How to test surprisal theory?

• Option 1. Test it on typical psycholinguistic experimental data, e.g. garden path sentences
Garden-pathing and surprisal

• Here’s another type of local syntactic ambiguity

When the dog scratched the vet and his new assistant removed the muzzle.

(Frazier & Rayner, 1982)
Garden-pathing and surprisal

• Here’s another type of local syntactic ambiguity

When the dog scratched the vet and his new assistant removed the muzzle.

(Frazier & Rayner, 1982)
Garden-pathing and surprisal

- Here’s another type of local syntactic ambiguity

When the dog scratched the vet and his new assistant removed the muzzle.

(Frazier & Rayner, 1982)
Garden-pathing and surprisal

- Here’s another type of local syntactic ambiguity

When the dog scratched the vet and his new assistant removed the muzzle.

(Frazier & Rayner, 1982)
Garden-pathing and surprisal

• Here’s another type of local syntactic ambiguity

When the dog scratched the vet and his new assistant \textcolor{cyan}{removed} the muzzle.

• Compare with:

When the dog scratched, the vet and his new assistant removed the muzzle.

When the dog scratched its owner the vet and his new assistant removed the muzzle.

(Frazier & Rayner, 1982)
Garden-pathing and surprisal

• Here’s another type of local syntactic ambiguity

When the dog scratched the vet and his new assistant removed the muzzle.

• Compare with:

When the dog scratched, the vet and his new assistant removed the muzzle.

When the dog scratched its owner the vet and his new assistant removed the muzzle.

(Frazier & Rayner, 1982)
Garden-pathing and surprisal

- Here’s another type of local syntactic ambiguity

When the dog scratched the vet and his new assistant removed the muzzle.

- Compare with:

When the dog scratched, the vet and his new assistant removed the muzzle.

When the dog scratched its owner the vet and his new assistant removed the muzzle.

(Frazier & Rayner, 1982)
Garden-pathing and surprisal

Here’s another type of local syntactic ambiguity.

When the dog scratched the vet and his new assistant removed the muzzle.

Compare with:

When the dog scratched, the vet and his new assistant removed the muzzle.

When the dog scratched its owner the vet and his new assistant removed the muzzle.

(Frazier & Rayner, 1982)
Garden-pathing and surprisal

• Here’s another type of local syntactic ambiguity

When the dog scratched the vet and his new assistant removed the muzzle.

difficulty here
(68ms/char)

easier
(50ms/char)

(Frazier & Rayner, 1982)

• Compare with:

When the dog scratched, the vet and his new assistant removed the muzzle.

When the dog scratched its owner the vet and his new assistant removed the muzzle.
How can we compute the surprisal of words in these sentences?

$$\text{Surprisal}(w_i) = -\log P(w_i | w_{1...i-1})$$

So we need a language model

Let’s use a Probabilistic Context Free Grammar (PCFG)!
A small PCFG for this sentence type

<table>
<thead>
<tr>
<th>Production</th>
<th>Probability</th>
<th>Nonterminal</th>
<th>Probability</th>
<th>Terminal</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>S → SBAR S</td>
<td>0.3</td>
<td>Conj</td>
<td>1</td>
<td>Adj</td>
<td>1</td>
</tr>
<tr>
<td>S → NP VP</td>
<td>0.7</td>
<td>Det</td>
<td>0.8</td>
<td>VP</td>
<td>0.5</td>
</tr>
<tr>
<td>SBAR → COMPL S</td>
<td>0.3</td>
<td>Det</td>
<td>0.1</td>
<td>VP</td>
<td>0.5</td>
</tr>
<tr>
<td>SBAR → COMPL S COMMA</td>
<td>0.7</td>
<td>Det</td>
<td>0.1</td>
<td>V</td>
<td>0.25</td>
</tr>
<tr>
<td>COMPL → When</td>
<td>1</td>
<td>N</td>
<td>0.2</td>
<td>V</td>
<td>0.25</td>
</tr>
<tr>
<td>NP → Det N</td>
<td>0.6</td>
<td>N</td>
<td>0.2</td>
<td>V</td>
<td>0.5</td>
</tr>
<tr>
<td>NP → Det Adj N</td>
<td>0.2</td>
<td>N</td>
<td>0.2</td>
<td>COMMA</td>
<td>1</td>
</tr>
<tr>
<td>NP → NP Conj NP</td>
<td>0.2</td>
<td>N</td>
<td>0.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>N</td>
<td>0.2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

We know how to calculate the probability of a tree from this PCFG, as the product of the derivations, e.g.:

\[ P(T) = P(S \rightarrow SBAR S) \times P(SBAR \rightarrow COMPL S) \times P(COMPL \rightarrow When) \times P(S \rightarrow NP VP) \times P(NP \rightarrow Det N) \times \cdots \times P(N \rightarrow assistant) \]

(analysis in Levy, 2011)
A small PCFG for this sentence type

- The probability of a sentence (aka. a *string*) $w_{1...n}$ is the sum of the probabilities of all trees that would yield that sentence
- In this small grammar, this sentence is globally unambiguous--there’s only one tree that’s compatible with the whole sentence
- In a larger grammar, ambiguity is common
- e.g. if there were two possible parses, we’d sum the probabilities of those two trees
### A small PCFG for this sentence type

| S  | → SBAR S | 0.3 | Conj → and | 1 | Adj → new | 1 |
| S  | → NP VP  | 0.7 | Det → the  | 0.8 | VP → V NP  | 0.5 |
| SBAR | → COMPL S | 0.3 | Det → its  | 0.1 | VP → V    | 0.5 |
| SBAR | → COMPL S COMMA | 0.7 | Det → his  | 0.1 | V → scratched | 0.25 |
| COMPL | → When  | 1 | N → dog    | 0.2 | V → removed | 0.25 |
| NP  | → Det N  | 0.6 | N → vet    | 0.2 | V → arrived | 0.5 |
| NP  | → Det Adj N | 0.2 | N → assistant | 0.2 | COMMA → ,  | 1 |
| NP  | → NP Conj NP | 0.2 | N → muzzle  | 0.2 | N → owner   | 0.2 |

- The probability of a sentence (aka. a *string*) $w_{1...n}$ is the sum of the probabilities of all trees that yield that sentence.
- The probability of a *sentence prefix* $w_{1...i}$ is the sum of the probabilities of all trees whose yield begins with $w_{1...i}$.
- But to calculate surprisal, we need the conditional probability
  
  $$P(w_i | w_{1...i-1}) = \frac{P(w_{1...i-1}, w_i)}{P(w_{1...i-1})} = \frac{P(w_{1...i})}{P(w_{1...i-1})}$$

- So how do we calculate these prefix probabilities?
Inference over infinite tree sets

Consider the following noun-phrase grammar:

\[
\begin{align*}
\frac{2}{3} & \quad \text{NP} \rightarrow \text{Det N} \\
\frac{1}{3} & \quad \text{NP} \rightarrow \text{NP PP} \\
1 & \quad \text{PP} \rightarrow \text{P NP}
\end{align*}
\]

\[
\begin{align*}
1 & \quad \text{Det} \rightarrow \text{the} \\
\frac{2}{3} & \quad \text{N} \rightarrow \text{dog} \\
\frac{1}{3} & \quad \text{N} \rightarrow \text{cat} \\
1 & \quad \text{P} \rightarrow \text{near}
\end{align*}
\]
Consider the following noun-phrase grammar:


definitions =
\begin{align*}
\frac{2}{3} & \rightarrow \text{Det N} \\
\frac{1}{3} & \rightarrow \text{NP PP} \\
\frac{1}{3} & \rightarrow \text{PP } P \text{ NP} \\
\frac{1}{3} & \rightarrow \text{N } P \\
\frac{1}{3} & \rightarrow \text{P } N \\
\frac{1}{3} & \rightarrow \text{the } \\
\frac{1}{3} & \rightarrow \text{dog } \\
\frac{1}{3} & \rightarrow \text{cat } \\
\end{align*}

Question: given a sentence starting with 

\textit{the} . . .

what is the probability that the next word is \textit{dog}?
Consider the following noun-phrase grammar:

\[
\begin{align*}
\frac{2}{3} & \quad \text{NP} \rightarrow \text{Det} \text{ N} \\
\frac{1}{3} & \quad \text{NP} \rightarrow \text{NP} \text{ PP} \\
1 & \quad \text{PP} \rightarrow \text{P} \text{ NP} \\
1 & \quad \text{Det} \rightarrow \text{the} \\
\frac{2}{3} & \quad \text{N} \rightarrow \text{dog} \\
\frac{1}{3} & \quad \text{N} \rightarrow \text{cat} \\
1 & \quad \text{P} \rightarrow \text{near}
\end{align*}
\]

Question: given a sentence starting with \textit{the}... what is the probability that the next word is \textit{dog}? Intuitively, the answers to this question should be

\[P(\text{dog}|\text{the}) = \frac{2}{3}\]
Inference over infinite tree sets

Consider the following noun-phrase grammar:

\[
\begin{align*}
&\frac{2}{3} \quad \text{NP} \to \text{Det N} \quad 1 \quad \text{Det} \to \text{the} \\
&\frac{1}{3} \quad \text{NP} \to \text{NP PP} \quad \frac{2}{3} \quad \text{N} \to \text{dog} \\
&\frac{1}{3} \quad \text{PP} \to \text{P NP} \quad \frac{1}{3} \quad \text{N} \to \text{cat} \\
&1 \quad \text{P} \to \text{near}
\end{align*}
\]

Question: given a sentence starting with \textit{the}... what is the probability that the next word is \textit{dog}?

Intuitively, the answers to this question should be

\[
P(\text{dog}|\text{the}) = \frac{2}{3}
\]

because the second word HAS to be either \textit{dog} or \textit{cat}.
Inference over infinite tree sets (2)

\[
\begin{align*}
& \frac{2}{3} \quad \text{NP} \rightarrow \text{Det N} \\
& \frac{1}{3} \quad \text{NP} \rightarrow \text{NP PP} \\
& 1 \quad \text{PP} \rightarrow \text{P NP} \\
& 1 \quad \text{Det} \rightarrow \text{the} \\
& \frac{2}{3} \quad \text{N} \rightarrow \text{dog} \\
& \frac{1}{3} \quad \text{N} \rightarrow \text{cat} \\
& 1 \quad \text{P} \rightarrow \text{near}
\end{align*}
\]

- We “should” just enumerate the trees that cover *the dog* . . . , and divide their total probability by that of *the* . . .
Inference over infinite tree sets (2)

\[
\begin{align*}
\frac{2}{3} & \quad \text{NP} \rightarrow \text{Det N} & \frac{2}{3} & \quad \text{Det} \rightarrow \text{the} \\
\frac{1}{3} & \quad \text{NP} \rightarrow \text{NP PP} & \frac{1}{3} & \quad \text{N} \rightarrow \text{dog} \\
1 & \quad \text{PP} \rightarrow \text{P NP} & 1 & \quad \text{N} \rightarrow \text{cat} \\
1 & \quad \text{P} \rightarrow \text{near} &
\end{align*}
\]

- We “should” just enumerate the trees that cover the dog . . ., and divide their total probability by that of the . . .
- . . .but there are infinitely many trees.
We “should” just enumerate the trees that cover *the dog* . . ., and divide their total probability by that of *the* . . .

. . . but there are infinitely many trees.
Shortcut 1: you can think of a *partial* tree as marginalizing over all completions of the partial tree. It has a corresponding marginal probability in the PCFG.
Shortcut 1: you can think of a *partial* tree as marginalizing over all completions of the partial tree. It has a corresponding marginal probability in the PCFG.
Shortcut 1: you can think of a *partial* tree as marginalizing over all completions of the partial tree.

It has a corresponding marginal probability in the PCFG.
Problem 2: there are still an infinite number of incomplete trees covering a partial input.
Problem 2: there are still an infinite number of incomplete trees covering a partial input.

BUT! These tree probabilities form a geometric series:

\[
P(\text{the dog ...}) = \frac{4}{9} + \frac{4}{27} + \frac{4}{81} + \frac{4}{243} + \cdots
\]
Problem 2: there are still an infinite number of incomplete trees covering a partial input.

\[
P(\text{the dog ...}) = \frac{4}{9} + \frac{4}{27} + \frac{4}{81} + \frac{4}{243} + \cdots = \frac{4}{9} \sum_{i=0}^{\infty} \left(\frac{1}{3}\right)^i
\]
Problem 2: there are still an infinite number of incomplete trees covering a partial input.

BUT! These tree probabilities form a geometric series:

\[ P(\text{the dog . . .}) = \frac{4}{9} + \frac{4}{27} + \frac{4}{81} + \frac{4}{243} + \cdots \]

\[ = \frac{4}{9} \sum_{i=0}^{\infty} \left(\frac{1}{3}\right)^i \]

\[ = \frac{2}{3} \]
Problem 2: there are still an infinite number of incomplete trees covering a partial input.

\[
P(\text{the dog . . .}) = \frac{4}{9} + \frac{4}{27} + \frac{4}{81} + \frac{4}{243} + \cdots
\]
\[
= \frac{4}{9} \sum_{i=0}^{\infty} \left(\frac{1}{3}\right)^i
\]
\[
= \frac{2}{3}
\]

\ldots which matches the original rule probability

\[
\frac{2}{3} N \rightarrow \text{dog}
\]
• Without going into all the details, we can generalize this process to compute prefix probabilities $P(w_{1...i})$ in polynomial time and space.

• Which allows us to compute $P(w_i|w_{1...i-1}) = \frac{P(w_{1...i})}{P(w_{1...i-1})}$.
A small PCFG for this sentence type

<table>
<thead>
<tr>
<th>Rule</th>
<th>Probability</th>
<th>Non-terminals</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>S → SBAR S</td>
<td>0.3</td>
<td>Conj → and</td>
<td>1</td>
</tr>
<tr>
<td>S → NP VP</td>
<td>0.7</td>
<td>Det → the</td>
<td>0.8</td>
</tr>
<tr>
<td>SBAR → COMPL S</td>
<td>0.3</td>
<td>Det → its</td>
<td>0.1</td>
</tr>
<tr>
<td>SBAR → COMPL S COMMA</td>
<td>0.7</td>
<td>Det → his</td>
<td>0.1</td>
</tr>
<tr>
<td>COMPL → When</td>
<td>1</td>
<td>N → dog</td>
<td>0.2</td>
</tr>
<tr>
<td>NP → Det N</td>
<td>0.6</td>
<td>N → vet</td>
<td>0.2</td>
</tr>
<tr>
<td>NP → Det Adj N</td>
<td>0.2</td>
<td>N → assistant</td>
<td>0.2</td>
</tr>
<tr>
<td>NP → NP Conj NP</td>
<td>0.2</td>
<td>N → muzzle</td>
<td>0.2</td>
</tr>
<tr>
<td>N → owner</td>
<td>0.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adj → new</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VP → V NP</td>
<td>0.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VP → V</td>
<td>0.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>V → scratched</td>
<td>0.25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>V → removed</td>
<td>0.25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>V → arrived</td>
<td>0.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>COMMA → ,</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

While the dog scratched the vet and his new assistant removed the muzzle

(analysis in Levy, 2011)
Two incremental trees
Two incremental trees

“Garden-path” analysis:

When the dog scratched the vet and his new assistant.
Two incremental trees

• “Garden-path” analysis:

When the dog scratched the vet and his new assistant
Two incremental trees

• “Garden-path” analysis:

\[ P(T | w_{1...10}) = 0.826 \]
Two incremental trees

- “Garden-path” analysis:

- Ultimately-correct analysis

\[ P(T|w_{1\ldots10}) = 0.826 \]
Two incremental trees

• “Garden-path” analysis:

\[ P(T|w_{1\ldots10}) = 0.826 \]

• Ultimately-correct analysis:

\[ P(T|w_{1\ldots10}) = 0.174 \]
Two incremental trees

- “Garden-path” analysis:

  \[ P(T|w_{1\ldots10}) = 0.826 \]

- Ultimately-correct analysis

  \[ P(T|w_{1\ldots10}) = 0.174 \]
Two incremental trees

• “Garden-path” analysis:

\[ P(T|w_{1\ldots10}) = 0.826 \]

• Ultimately-correct analysis

\[ P(T|w_{1\ldots10}) = 0.174 \]
Two incremental trees

- “Garden-path” analysis:

Disambiguating word probability marginalizes over incremental trees:

\[ P(T|w_{1...10}) = 0.826 \]

- Ultimately-correct analysis

\[ P(T|w_{1...10}) = 0.174 \]
Preceding context can disambiguate

- "its owner" takes up the object slot of scratched

<table>
<thead>
<tr>
<th>Condition</th>
<th>Surprisal at Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>NP absent</td>
<td>4.2</td>
</tr>
<tr>
<td>NP present</td>
<td>2</td>
</tr>
</tbody>
</table>
Sensitivity to verb argument structure

- A superficially similar example:

When the dog arrived the vet and his new assistant removed the muzzle.

Easier here (Staub, 2007)
Sensitivity to verb argument structure

• A superficially similar example:

When the dog arrived the vet and his new assistant removed the muzzle.

But harder here!  Easier here

(Staub, 2007)
Sensitivity to verb argument structure

• A superficially similar example:

When the dog arrived the vet and his new assistant removed the muzzle.

(c.f. When the dog scratched the vet and his new assistant removed the muzzle.)
## Modeling argument-structure sensitivity

<table>
<thead>
<tr>
<th>Rule</th>
<th>Probability</th>
<th>Action 1</th>
<th>Probability</th>
<th>Action 2</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>S → SBAR S</td>
<td>0.3</td>
<td>Conj → and</td>
<td>1</td>
<td>Adj → new</td>
<td>1</td>
</tr>
<tr>
<td>S → NP VP</td>
<td>0.7</td>
<td>Det → the</td>
<td>0.8</td>
<td>VP → V NP</td>
<td>0.5</td>
</tr>
<tr>
<td>SBAR → COMPL S</td>
<td>0.3</td>
<td>Det → its</td>
<td>0.1</td>
<td>VP → V</td>
<td>0.5</td>
</tr>
<tr>
<td>SBAR → COMPL S COMMA</td>
<td>0.7</td>
<td>Det → his</td>
<td>0.1</td>
<td>V → scratched</td>
<td>0.25</td>
</tr>
<tr>
<td>COMPL → When</td>
<td>1</td>
<td>N → dog</td>
<td>0.2</td>
<td>V → removed</td>
<td>0.25</td>
</tr>
<tr>
<td>NP → Det N</td>
<td>0.6</td>
<td>N → vet</td>
<td>0.2</td>
<td>V → arrived</td>
<td>0.5</td>
</tr>
<tr>
<td>NP → Det Adj N</td>
<td>0.2</td>
<td>N → assistant</td>
<td>0.2</td>
<td>COMMA → ,</td>
<td>1</td>
</tr>
<tr>
<td>NP → NP Conj NP</td>
<td>0.2</td>
<td>N → muzzle</td>
<td>0.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>N → owner</td>
<td>0.2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Modeling argument-structure sensitivity

<table>
<thead>
<tr>
<th>S</th>
<th>→ SBAR S</th>
<th>0.3</th>
<th>Conj → and</th>
<th>1</th>
<th>Adj → new</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>→ NP VP</td>
<td>0.7</td>
<td>Det → the</td>
<td>0.8</td>
<td>VP → V NP</td>
<td>0.5</td>
</tr>
<tr>
<td>SBAR → COMPL S</td>
<td>0.3</td>
<td>Det → its</td>
<td>0.1</td>
<td>VP → V</td>
<td>0.5</td>
<td></td>
</tr>
<tr>
<td>SBAR → COMPL S COMMA</td>
<td>0.7</td>
<td>Det → his</td>
<td>0.1</td>
<td>V → scratched</td>
<td>0.25</td>
<td></td>
</tr>
<tr>
<td>COMPL → When</td>
<td>1</td>
<td>N → dog</td>
<td>0.2</td>
<td>V → removed</td>
<td>0.25</td>
<td></td>
</tr>
<tr>
<td>NP → Det N</td>
<td>0.6</td>
<td>N → vet</td>
<td>0.2</td>
<td>V → arrived</td>
<td>0.5</td>
<td></td>
</tr>
<tr>
<td>NP → Det Adj N</td>
<td>0.2</td>
<td>N → assistant</td>
<td>0.2</td>
<td>COMMA → ,</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>NP → NP Conj NP</td>
<td>0.2</td>
<td>N → muzzle</td>
<td>0.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>N → owner</td>
<td>0.2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| VP → V NP | 0.5 | VP → Vtrans NP | 0.45 |
| VP → V   | 0.5 | VP → Vtrans   | 0.05 |
| V → scratched | 0.25 | VP → Vintrans | 0.45 |
| V → removed | 0.25 | VP → Vintrans NP | 0.05 |
| V → arrived | 0.5 | Vtrans → scratched | 0.5 |
|            |       | Vtrans → removed | 0.5 |
|            |       | Vintrans → arrived | 1 |

(Johnson, 1999; Klein & Manning, 2003)
When the dog arrived the vet and his new assistant removed the muzzle.

When the dog scratched the vet and his new assistant removed the muzzle.

Transitivity-distinguishing PCFG

<table>
<thead>
<tr>
<th>Condition</th>
<th>Ambiguity onset</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intransitive (arrived)</td>
<td>2.11</td>
<td>3.20</td>
</tr>
<tr>
<td>Transitive (scratched)</td>
<td>0.44</td>
<td>8.04</td>
</tr>
</tbody>
</table>
Move to broad coverage

- Instead of the pedagogical grammar, a “broad-coverage” grammar from the parsed Brown corpus (11,984 rules)
- Relative-frequency estimation of rule probabilities (“vanilla” PCFG)
How to test surprisal theory?

• Option 1. Test it on typical psycholinguistic experimental data, e.g. garden path sentences
  • ✓
    • Demonstrates that surprisal predicts specific, known psycholinguistic effects

• Disadvantages:
  • Psycholinguistic stimuli may not be representative of naturalistic language use.
  • Surprisal makes broad-coverage quantitative predictions, so it would be nice to test those more explicitly
How to test surprisal theory?

• Option 2: Test its broad-coverage predictions

• Dundee Corpus: Eye-tracking corpus
  • 10 participants
  • each read 51,000 words of English newspaper text

• Demberg & Keller (2008) tested surprisal theory predictions for RTs on *every word* in the corpus
  • Trained a PCFG on a different English corpus to make surprisal predictions

• Using regression analysis, they found that surprisal was a significant predictor of reading times in the corpus as a whole ✓

• Disadvantages: This is still not a strong test of the *shape* of the surprisal effect
Recall...

• Surprisal($w_i$) = $-\log P(w_i|\text{CONTEXT})$

• Claim: Reading time($w_i$) $\propto$ Surprisal($w_i$)

• However, this isn’t the only possible shape for the effect of predictability on RTs, or even the only shape that has been proposed in the literature

(Hale, 2001; Levy, 2008)
Proposed relationships between predictability and reading time

(from Smith & Levy, 2013)
Estimating probability/time curve shape

(from Smith & Levy, 2013)
Estimating probability/time curve shape

- As a proxy for “processing difficulty,” reading time in two different methods: self-paced reading & eye-tracking

(from Smith & Levy, 2013)
Estimating probability/time curve shape

- As a proxy for “processing difficulty,” reading time in two different methods: self-paced reading & eye-tracking
- Challenge: we need big data to estimate curve shape, but probability correlated with confounding variables

(from Smith & Levy, 2013)
Estimating probability/time curve shape

- As a proxy for “processing difficulty,” reading time in two different methods: self-paced reading & eye-tracking
- Challenge: we need big data to estimate curve shape, but probability correlated with confounding variables

Brown data availability
(self-paced reading)

Dundee data availability
(eye-tracking)

(5K words)

(50K words)
Estimating probability/time curve shape

• Method:
  • Estimate surprisal of each word using trigram model
    • The best broad-coverage word probability model at the time
    • (Recall, any language model can be used to compute surprisal, not just PCFGs)
  • Estimate how well surprisal predicts RTs (over and above frequency and other confounding variables) and the shape of the effect using Generalized Additive Model (GAM) regression
    • Can control for the effects of multiple predictor variables
    • Doesn’t assume a linear relationship between the independent and dependent variables, so it can be used to estimate the shape of an effect

(from Smith & Levy, 2013)
Estimating probability/time curve shape

- GAM regression: total contribution of word (trigram) probability to RT near-linear over 6 orders of magnitude!

(Smith & Levy, 2013)
How to test surprisal theory?

• Option 1. Test it on typical psycholinguistic experimental data, e.g. garden path sentences  
  • ✓

• Option 2: Test its broad-coverage predictions  
  • ✓

• Option 3: Explicitly test the predicted shape of the predictability effect on RTs  
  • ✓
Processing difficulty summary

• Two potential sources of processing difficulty
  • Memory limitations
  • Expectation-based

• Evidence for both

• Expectation-based difficulty can be quantified via surprisal
  • Receives empirical support across many studies & methods of testing

• Surprisal theory claims that the “work” of sentence processing is updating distributions over incremental parses
  • Predicting the next word is implicit in calculating probabilities over sentence structures
  • Processing a possible tree structure (including its syntactic structure) “bottoms out” in the probability of processing each word incrementally