Computational Psycholinguistics

Integrating NLP modeling and experimental psycholinguistics to investigate real-time human language use

Roger Levy    Klington Bicknell    Nathaniel Smith
Departments of Linguistics & Cognitive Science
University of California – San Diego

NAACL-HLT 2010 tutorial

Slides available at http://grammar.ucsd.edu/cpl/naacl2010tutorial.html
Basic desiderata

• Realistic models of human sentence processing must account for
  • Robustness to arbitrary input
  • Accurate disambiguation
  • Inference on basis of incomplete input
• Processing difficulty is differential and localized
Incrementality and Rationality

- **Ambiguity** a major problem in language understanding
- **Parsing**: sentence $w \rightarrow$ distribution on structures $P(T|w)$
- **Incremental parsing**: for a partial sentence $w_{1\ldots i}$, assign a distribution over partial structures that implies $P(T|w_{1\ldots i})$
- Lots of evidence that people use diverse information sources to perform incremental parsing rationally

“The boy will eat…”

- Today: give you an intro to the computational & empirical picture
Contents of the day’s talk

- Review of exact incremental inference techniques for PCFGs
- Overview of key results in modeling ambiguity resolution and expectation-based facilitation in online comprehension
- Coffee Break
- Rational, probabilistic inference & speaker choice in language production
- Cognitive limitations and approximate inference in online comprehension
- Additional theoretical challenges, bounded rationality, input uncertainty
- Summary, open issues, future directions, questions
PCFGs and the Earley Algorithm

Computational Psycholinguistics Tutorial
NAACL 2010
Context-free Grammars

A context-free grammar (CFG) consists of a tuple $(N, V, S, R)$ such that:

- $N$ is a finite set of non-terminal symbols;
- $V$ is a finite set of terminal symbols;
- $S$ is the start symbol;
- $R$ is a finite set of rules of the form $X \rightarrow \alpha$ where $X \in N$ and $\alpha$ is a sequence of symbols drawn from $N \cup V$.

A CFG derivation is the recursive expansion of non-terminal symbols into a string of terminal symbols by rules in $R$, starting with $S$, and a derivation tree $T$ is the history of those rule applications.
Let our grammar (the rule-set $R$) be

- $S \rightarrow NP\ VP$
- $NP \rightarrow \text{Det}\ N$
- $NP \rightarrow NP\ PP$
- $PP \rightarrow P\ NP$
- $VP \rightarrow V$

Det $\rightarrow$ the
N $\rightarrow$ dog
N $\rightarrow$ cat
P $\rightarrow$ near
V $\rightarrow$ growled

The nonterminal set $N$ is $\{S, NP, VP, Det, N, P, V\}$, the terminal set $V$ is $\{\text{the, dog, cat, near, growled}\}$, and our start symbol $S$ is $S$. 
Context-free Grammars: an example II

S → NP VP
NP → Det N
NP → NP PP
PP → P NP
VP → V

Det → the
N → dog
N → cat
P → near
V → growled

Here is a *derivation* and the resulting *derivation tree*:

```
S
```
Context-free Grammars: an example II

S → NP VP
NP → Det N
NP → NP PP
PP → P NP
VP → V

Det → the
N → dog
N → cat
P → near
V → growled

Here is a *derivation* and the resulting *derivation tree*:
Context-free Grammars: an example II

S → NP VP
NP → Det N
NP → NP PP
PP → P NP
VP → V

Det → the
N → dog
N → cat
P → near
V → growled

Here is a derivation and the resulting derivation tree:
Context-free Grammars: an example II

S → NP VP
NP → Det N
NP → NP PP
PP → P NP
VP → V

Det → the
N → dog
N → cat
P → near
V → growled

Here is a derivation and the resulting derivation tree:
Context-free Grammars: an example II

\[
\begin{align*}
S & \rightarrow NP \ VP \\
NP & \rightarrow \text{Det} \ N \\
NP & \rightarrow NP \ PP \\
PP & \rightarrow P \ NP \\
VP & \rightarrow V
\end{align*}
\]

Here is a *derivation* and the resulting *derivation tree*:

```
the dog growled near
```

```
the

S

NP

NP

NP

PP

Det

V

```

```
the

VP

```

```
PP

```

```
P

```

```
N

```

```
V

```

```
the dog growled near

```
Context-free Grammars: an example II

S → NP VP
NP → Det N
NP → NP PP
PP → P NP
VP → V

Det → the
N → dog
N → cat
P → near
V → growled

Here is a derivation and the resulting derivation tree:
Context-free Grammars: an example II

S → NP VP
NP → Det N
NP → NP PP
PP → P NP
VP → V

Det → the
N → dog
N → cat
P → near
V → growled

Here is a *derivation* and the resulting *derivation tree*:
Context-free Grammars: an example II

\[
\begin{align*}
S & \rightarrow NP \ VP \\
NP & \rightarrow \text{Det} \ N \\
NP & \rightarrow NP \ PP \\
PP & \rightarrow P \ NP \\
VP & \rightarrow V
\end{align*}
\]

Det\rightarrow the \\
N \rightarrow dog \\
N \rightarrow cat \\
P \rightarrow near \\
V \rightarrow growled

Here is a *derivation* and the resulting *derivation tree*:

```
the
dog
near
```
Context-free Grammars: an example II

S → NP VP
NP → Det N
NP → NP PP
PP → P NP
VP → V

Det → the
N → dog
N → cat
P → near
V → growled

Here is a *derivation* and the resulting *derivation tree*:

```
S
  NP
   NP
    Det N
     the dog
    PP
     P near
     Det N
```

```
Context-free Grammars: an example II

S → NP VP
NP → Det N
NP → NP PP
PP → P NP
VP → V

Det → the
N → dog
N → cat
P → near
V → growled

Here is a derivation and the resulting derivation tree:
Context-free Grammars: an example II

\[
\begin{align*}
S & \rightarrow NP \ VP \\
NP & \rightarrow Det \ N \\
NP & \rightarrow NP \ PP \\
PP & \rightarrow P \ NP \\
VP & \rightarrow V \\
Det & \rightarrow the \\
N & \rightarrow dog \\
N & \rightarrow cat \\
P & \rightarrow near \\
V & \rightarrow growled
\end{align*}
\]

Here is a *derivation* and the resulting *derivation tree*:
Context-free Grammars: an example II

\[
S \to NP \ VP \\
NP \to Det \ N \\
NP \to NP \ PP \\
PP \to P \ NP \\
VP \to V
\]

\[
\begin{array}{ll}
\text{Det} & \to \text{the} \\
\text{N} & \to \text{dog} \\
\text{N} & \to \text{cat} \\
\text{P} & \to \text{near} \\
\text{V} & \to \text{growled}
\end{array}
\]

Here is a derivation and the resulting derivation tree:
Context-free Grammars: an example II

S → NP VP
NP → Det N
NP → NP PP
PP → P NP
VP → V

Det → the
N → dog
N → cat
P → near
V → growled

Here is a derivation and the resulting derivation tree:
A probabilistic context-free grammar (PCFG) consists of a tuple $(N, V, S, R, P)$ such that:

- $N$ is a finite set of non-terminal symbols;
- $V$ is a finite set of terminal symbols;
- $S$ is the start symbol;
- $R$ is a finite set of rules of the form $X \rightarrow \alpha$ where $X \in N$ and $\alpha$ is a sequence of symbols drawn from $N \cup V$;
- $P$ is a mapping from $R$ into probabilities, such that for each $X \in N$,

$$\sum_{[X \rightarrow \alpha] \in R} P(X \rightarrow \alpha) = 1$$

The probability $P(T)$ of a derivation tree is simply the product of the probabilities of each rule application.
Example PCFG

1  S  → NP VP
0.8 NP  → Det N
0.2 NP  → NP PP
1  PP  → P NP
1  VP  → V

1  Det  → the
0.5  N  → dog
0.5  N  → cat
1  P  → near
1  V  → growled

\[
P(T) = 1 \times 0.2 \times 0.8 \times 1 \times 0.5 \times 0.8 \times 1 \times \times 0.8 \times 1 \times 0.5 \times 1 \times 1
\]
\[
= 0.032
\]
We just learned how to calculate the *probability of a tree*

The *probability of a string* $w_1 \ldots n$ is the sum of the probabilities of all trees whose yield is $w_1 \ldots n$

The *probability of a string prefix* $w_1 \ldots i$ is the sum of the probabilities of all trees whose yield *begins with* $w_1 \ldots i$

If we had the probabilities of two string prefixes $w_1 \ldots i−1$ and $w_1 \ldots i$, we could calculate the conditional probability $P(w_i|w_1 \ldots i−1)$ as their ratio:

$$P(w_i|w_1 \ldots i−1) = \frac{P(w_1 \ldots i)}{P(w_1 \ldots i−1)}$$
Inference over infinite tree sets

Consider the following noun-phrase grammar:

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

\[
\begin{align*}
1 & \quad \text{Det} \rightarrow \text{the} \\
\frac{2}{3} & \quad \text{N} \rightarrow \text{dog} \\
\frac{1}{3} & \quad \text{N} \rightarrow \text{cat} \\
1 & \quad \text{P} \rightarrow \text{near}
\end{align*}
\]
Inference over infinite tree sets

Consider the following noun-phrase grammar:

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

1 \quad \text{Det} \rightarrow \text{the}

\begin{align*}
\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}?
Consider the following noun-phrase grammar:

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

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}

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:

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

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

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

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

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

We “should” just enumerate the trees that cover *the dog* . . . ,
Inference over infinite tree sets (2)

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

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

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.
Inference over infinite tree sets (2)

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 \textit{partial} tree as marginalizing over all completions of the partial tree.
It has a corresponding marginal probability in the PCFG.

\[
\begin{align*}
\text{Det} \rightarrow \text{the} \\
\text{N} \rightarrow \text{dog} \\
\text{N} \rightarrow \text{cat} \\
\text{P} \rightarrow \text{near}
\end{align*}
\]
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:

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

BUT! These tree probabilities form a geometric series:

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

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

\[
= \frac{4}{9} \cdot \frac{1}{1 - \frac{1}{3}} = \frac{2}{3}
\]

\[
\frac{2}{3} N \rightarrow \text{dog}
\]
Generalizing the geometric series induced by rule recursion

In general, these infinite tree sets arise due to *left recursion* in a probabilistic grammar

\[
A \rightarrow B \alpha \quad \quad \quad \quad \quad \quad \quad B \rightarrow A \beta
\]

*(Stolcke, 1995)*
Generalizing the geometric series induced by rule recursion

In general, these infinite tree sets arise due to *left recursion* in a probabilistic grammar

\[
A \rightarrow B \alpha \\
B \rightarrow A \beta
\]

We can formulate a stochastic *left-corner matrix* of transitions between categories:

\[
P_L = \begin{bmatrix}
A & B & \cdots & K \\
A & 0.3 & 0.7 & \cdots & 0 \\
B & 0.1 & 0.1 & \cdots & 0.2 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
K & 0.2 & 0.1 & \cdots & 0.2
\end{bmatrix}
\]

*(Stolcke, 1995)*
Generalizing the geometric series induced by rule recursion

In general, these infinite tree sets arise due to *left recursion* in a probabilistic grammar

\[ A \rightarrow B \alpha \quad \text{and} \quad B \rightarrow A \beta \]

We can formulate a stochastic *left-corner matrix* of transitions between categories:

\[
P_L = \begin{bmatrix}
A & B & \cdots & K \\
A & 0.3 & 0.7 & \cdots & 0 \\
B & 0.1 & 0.1 & \cdots & 0.2 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
K & 0.2 & 0.1 & \cdots & 0.2 \\
\end{bmatrix}
\]

and solve for its closure \( R_L = (I - P_L)^{-1} \).

(Stolcke, 1995)
Generalizing the geometric series

The closure of our left-corner matrix is

\[
R_L = \begin{pmatrix}
1 & \frac{3}{2} & 0 & 1 & 0 & 0 & 0 \\
0 & \frac{3}{2} & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 \\
\end{pmatrix}
\]
Generalizing the geometric series

<table>
<thead>
<tr>
<th></th>
<th>ROOT → NP</th>
<th>1</th>
<th></th>
<th>Det → the</th>
</tr>
</thead>
<tbody>
<tr>
<td>2³</td>
<td>NP → Det N</td>
<td>2²</td>
<td>3</td>
<td>N → dog</td>
</tr>
<tr>
<td>1³</td>
<td>NP → NP PP</td>
<td>3²</td>
<td>3</td>
<td>N → cat</td>
</tr>
<tr>
<td>1</td>
<td>PP → P NP</td>
<td>1</td>
<td></td>
<td>P → near</td>
</tr>
</tbody>
</table>

► The closure of our left-corner matrix is

$$R_L = \begin{pmatrix}
1 & 3^2 & 0 & 1 & 0 & 0 & 0 \\
0 & 2^3 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}$$

► Refer to an entry \((X, Y)\) in this matrix as \(R(X \Rightarrow_L Y)\)
Generalizing the geometric series

1. \( \text{ROOT} \rightarrow \text{NP} \)

2. \( \frac{2}{3} \) \( \text{NP} \rightarrow \text{Det N} \)

3. \( \frac{1}{3} \) \( \text{NP} \rightarrow \text{NP PP} \)

4. \( \frac{1}{3} \) \( \text{PP} \rightarrow \text{P NP} \)

1. \( \text{Det} \rightarrow \text{the} \)

2. \( \frac{2}{3} \) \( \text{N} \rightarrow \text{dog} \)

3. \( \frac{1}{3} \) \( \text{N} \rightarrow \text{cat} \)

1. \( \text{P} \rightarrow \text{near} \)

The closure of our left-corner matrix is

\[
R_L = \begin{pmatrix}
1 & \frac{3}{2} & 0 & 1 & 0 & 0 & 0 \\
0 & \frac{3}{2} & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & \\
0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 
\end{pmatrix}
\]

Refer to an entry \((X, Y)\) in this matrix as \(R(X \Rightarrow^*_L Y)\)

Note that the \(\frac{3}{2}\) “bonus” accrued for left-recursion of NPs appears in the (ROOT,NP) and (NP,NP) cells of the matrix.
Generalizing the geometric series

\[
\begin{array}{c|c}
1 & \text{ROOT} \rightarrow \text{NP} \\
2 & \text{NP} \rightarrow \text{Det N} \\
2 & \text{NP} \rightarrow \text{NP PP} \\
1 & \text{PP} \rightarrow \text{P NP} \\
1 & \text{Det} \rightarrow \text{the} \\
2 & \text{N} \rightarrow \text{dog} \\
2 & \text{N} \rightarrow \text{cat} \\
1 & \text{P} \rightarrow \text{near}
\end{array}
\]

- The closure of our left-corner matrix is

\[
R_L = \begin{pmatrix}
1 & 3 & 0 & 1 & 0 & 0 & 0 \\
0 & 3 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}
\]

- Refer to an entry \((X, Y)\) in this matrix as \(R(X \Rightarrow_L Y)\)

- Note that the \(\frac{3}{2}\) “bonus” accrued for left-recursion of NPs appears in the \((\text{ROOT, NP})\) and \((\text{NP, NP})\) cells of the matrix.

- We need to do the same with unary chains, constructing a unary-closure matrix \(R_U\).
Efficient incremental parsing: the probabilistic Earley algorithm

We can use the Earley algorithm (Earley, 1970) in a probabilistic incarnation (Stolcke, 1995) to deal with these infinite tree sets.

The (slightly oversimplified) probabilistic Earley algorithm has two fundamental types of operations:

- **Prediction**: if $Y$ is a possible goal, and $Y$ can lead to $Z$ through a left corner, choose a rule $Z \rightarrow \alpha$ and set up $\alpha$ as a new sequence of possible goals.

- **Completion**: if $Y$ is a possible goal, $Y$ can lead to $Z$ through unary rewrites, and we encounter a completed $Z$, absorb it and move on to the next sub-goal in the sequence.
Efficient incremental parsing: the probabilistic Earley algorithm

- Parsing consists of constructing a chart of states (items)
- A state has the following structure:

\[ X \rightarrow \alpha \circ \beta \]

- Forward probability
- Inside probability

- The forward probability is the total probability of getting from the root at the start of the sentence through to this state
- The inside probability is the “bottom-up” probability of the state
Efficient incremental parsing: the probabilistic Earley algorithm

Inference rules for probabilistic Earley:

- **Prediction:**
  \[
  \frac{X \rightarrow \beta \circ Y \gamma}{p \quad q} \quad a : R(Y \Rightarrow^*_L Z) \quad b : Z \rightarrow \alpha
  \]
  \[
  \frac{Z \rightarrow \circ \alpha}{abp \quad b}
  \]
Efficient incremental parsing: the probabilistic Earley algorithm

Inference rules for probabilistic Earley:

- **Prediction:**
  \[
  X \rightarrow \beta \circ Y \gamma \\
  p \quad q \\
  \]

  \[
  a : R( Y \Rightarrow_{L} Z ) \quad b : Z \rightarrow \alpha \\
  \\
  Z \rightarrow \circ \alpha \\
  abp \quad b
  \]

- **Completion:**
  \[
  X \rightarrow \beta \circ Y \gamma \\
  p \quad q \\
  \]

  \[
  a : R( Y \Rightarrow_{U} Z ) \quad Z \rightarrow \alpha \circ \\
  Z \rightarrow \circ \alpha \\
  \\
  b \quad c
  \]

  \[
  X \rightarrow \beta Y \circ \gamma \\
  acp \quad acq
  \]
Efficient incremental parsing: probabilistic Earley
Efficient incremental parsing: probabilistic Earley
Efficient incremental parsing: probabilistic Earley

Det → o the
1 1

NP → o Det N
\( \frac{2}{3} \times \frac{3}{2} \quad \frac{2}{3} \)

NP → o NP PP
\( \frac{1}{3} \times \frac{3}{2} \quad \frac{1}{3} \)

ROOT → o NP
1 1

do the near the
Efficient incremental parsing: probabilistic Earley

Det → ◦ the
1  1

NP → ◦ Det N
\( \frac{2}{3} \times \frac{3}{2} \times \frac{2}{3} \)

NP → ◦ NP PP
\( \frac{1}{3} \times \frac{3}{2} \times \frac{1}{3} \)

ROOT → ◦ NP
1  1
Efficient incremental parsing: probabilistic Earley

\[
\text{Det} \rightarrow \circ \text{the} \\
1 \quad 1
\]

\[
\text{NP} \rightarrow \circ \text{Det} \ N \\
\frac{2}{3} \times \frac{3}{2} \quad \frac{2}{3}
\]

\[
\text{NP} \rightarrow \circ \text{NP} \ PP \\
\frac{1}{3} \times \frac{3}{2} \quad \frac{1}{3}
\]

\[
\text{ROOT} \rightarrow \circ \text{NP} \\
1 \quad 1
\]

\[
\text{NP} \rightarrow \text{Det} \circ \text{N} \\
1 \quad \frac{2}{3}
\]

\[
\text{Det} \rightarrow \circ \text{the} \\
1 \quad 1
\]

the \quad dog \quad near \quad the
Efficient incremental parsing: probabilistic Earley

Det → \text{o}the
1 \hspace{1cm} 1

NP → \text{o}Det N
\frac{2}{3} \times \frac{3}{2} \hspace{1cm} \frac{2}{3}

NP → \text{o}NP PP
\frac{1}{3} \times \frac{3}{2} \hspace{1cm} \frac{1}{3}

ROOT → \text{o}NP
1 \hspace{1cm} 1

\text{the} \hspace{1cm} \text{dog} \hspace{1cm} \text{near} \hspace{1cm} \text{the}
Efficient incremental parsing: probabilistic Earley

Det → ◦ the
1 1

NP → ◦ Det N
\( \frac{2}{3} \times \frac{3}{2} \quad \frac{2}{3} \)

NP → ◦ NP PP
\( \frac{1}{3} \times \frac{3}{2} \quad \frac{1}{3} \)

ROOT → ◦ NP
\( \frac{1}{3} \quad \frac{1}{3} \)

N → ◦ cat
\( \frac{1}{3} \quad \frac{1}{3} \)

N → ◦ dog
\( \frac{2}{3} \quad \frac{2}{3} \)

NP → Det ◦ N
1 \( \frac{2}{3} \)

Det → ◦ the
1 1

the dog near the
Efficient incremental parsing: probabilistic Earley

\[
\text{Det} \rightarrow \circ \text{the} \\
1 \\
\text{NP} \rightarrow \circ \text{Det N} \\
\frac{2}{3} \times \frac{3}{2} \\
\text{NP} \rightarrow \circ \text{NP PP} \\
\frac{1}{3} \times \frac{3}{2} \times \frac{2}{3} \\
\text{ROOT} \rightarrow \circ \text{NP} \\
\frac{1}{3} \times \frac{3}{2} \\
\text{Det} \rightarrow \circ \text{the} \\
1 \\
\text{NP} \rightarrow \circ \text{Det N} \\
\frac{2}{3} \\
\text{N} \rightarrow \circ \text{cat} \\
\frac{1}{3} \times \frac{1}{3} \\
\text{N} \rightarrow \circ \text{dog} \\
\frac{2}{3} \times \frac{2}{3} \\
\text{NP} \rightarrow \circ \text{Det N} \\
1 \\
\text{the} \\
\text{dog} \\
\text{near} \\
\text{the}
Efficient incremental parsing: probabilistic Earley
Efficient incremental parsing: probabilistic Earley

Det → ◦ the
1 1

NP → ◦ Det N
\frac{2}{3} \times \frac{3}{2} \frac{2}{3}

NP → ◦ NP PP
\frac{1}{3} \times \frac{3}{2} \frac{1}{3}

ROOT → ◦ NP
1 1

N → ◦ cat
\frac{1}{3} \frac{1}{3}

N → ◦ dog
\frac{2}{3} \frac{2}{3}

NP → Det ◦ N
1 \frac{2}{3}

Det → the ◦
1 1

N → dog ◦
\frac{2}{3} \frac{2}{3}

dog

near

the
Efficient incremental parsing: probabilistic Earley

\[
\begin{align*}
\text{ROOT} & \rightarrow \text{NP} \circ \\
& \quad \frac{4}{9} \quad \frac{4}{9} \\
\text{Det} & \rightarrow \text{the} \\
& \quad 1 \quad 1 \\
\text{NP} & \rightarrow \text{Det} \circ N \\
& \quad \frac{2}{3} \times \frac{3}{2} \quad \frac{2}{3} \\
\text{NP} & \rightarrow \text{NP} \circ \text{Det} \circ N \\
& \quad \frac{1}{3} \times \frac{3}{2} \quad \frac{1}{3} \\
\text{ROOT} & \rightarrow \text{NP} \\
& \quad 1 \quad 1 \\
\text{NP} & \rightarrow \text{Det} \circ N \\
& \quad 1 \quad \frac{2}{3} \\
\text{Det} & \rightarrow \text{the} \\
& \quad 1 \quad 1 \\
\text{NP} & \rightarrow \text{Det} \circ N \\
& \quad \frac{2}{3} \quad \frac{2}{3} \\
\text{N} & \rightarrow \text{cat} \\
& \quad \frac{1}{3} \quad \frac{1}{3} \\
\text{N} & \rightarrow \text{dog} \\
& \quad \frac{2}{3} \quad \frac{2}{3} \\
\text{N} & \rightarrow \text{dog} \\
& \quad \frac{2}{3} \quad \frac{2}{3} \\
\text{the} & \\
& \\
\text{dog} & \\
& \\
\text{near} & \\
& \\
\text{the} & \\
& \\
\end{align*}
\]
Efficient incremental parsing: probabilistic Earley

\[
\begin{align*}
\text{Det} & \rightarrow \text{the} \\
\frac{2}{3} & \times \frac{1}{3} \\
\text{NP} & \rightarrow \text{Det N} \\
\frac{2}{3} & \times \frac{2}{3} \\
\text{NP} & \rightarrow \text{NP PP} \\
\frac{1}{3} & \times \frac{3}{2} \\
\text{ROOT} & \rightarrow \text{NP} \\
1 & 1
\end{align*}
\]
Efficient incremental parsing: probabilistic Earley

- **ROOT → NP**
  - \( \frac{4}{9} \times \frac{4}{9} \)

- **Det → ◦ the**
  - \( \frac{1}{3} \times \frac{3}{2} \times \frac{2}{3} \)

- **NP → ◦ Det N**
  - \( \frac{2}{3} \times \frac{3}{2} \times \frac{2}{3} \)

- **NP → ◦ NP PP**
  - \( \frac{1}{3} \times \frac{3}{2} \times \frac{1}{3} \)

- **ROOT → ◦ NP**
  - \( \frac{1}{3} \times \frac{3}{2} \times \frac{1}{3} \)

- **P → ◦ near**
  - \( \frac{2}{9} \times 1 \)

- **PP → ◦ P NP**
  - \( \frac{2}{9} \times 1 \)

- **NP → ◦ Det N**
  - \( \frac{2}{3} \times \frac{3}{2} \times \frac{2}{3} \)

- **Det → ◦ the**
  - \( \frac{1}{3} \times \frac{3}{2} \times \frac{2}{3} \)

- **NP → ◦ Det N**
  - \( \frac{2}{3} \times \frac{3}{2} \times \frac{2}{3} \)

- **N → ◦ cat**
  - \( \frac{1}{3} \times \frac{3}{2} \times \frac{1}{3} \)

- **N → ◦ dog**
  - \( \frac{2}{3} \times \frac{3}{2} \times \frac{2}{3} \)

- **N → ◦ dog**
  - \( \frac{2}{3} \times \frac{3}{2} \times \frac{2}{3} \)

- **the**

- **dog**

- **near**

- **the**
Efficient incremental parsing: probabilistic Earley

\[
\begin{align*}
\text{Efficient incremental parsing: probabilistic Earley} &\quad \text{ROOT} \rightarrow \text{NP} \circ \text{NP} \rightarrow \text{NP} \circ \text{PP} \\
\text{Det} &\rightarrow \text{the} \\
\text{NP} &\rightarrow \text{Det} \circ \text{N} \\
\text{NP} &\rightarrow \text{NP} \circ \text{NP} \circ \text{PP} \\
\text{ROOT} &\rightarrow \text{NP} \\
\text{NP} &\rightarrow \text{Det} \circ \text{N} \\
\text{Det} &\rightarrow \text{the} \\
\text{N} &\rightarrow \text{dog} \\
\text{NP} &\rightarrow \text{Det} \circ \text{N} \\
\text{P} &\rightarrow \text{near} \\
\text{PP} &\rightarrow \text{P} \circ \text{NP} \\
\text{the} &\quad \text{dog} \\
\text{near} &\quad \text{the}
\end{align*}
\]
Efficient incremental parsing: probabilistic Earley
Efficient incremental parsing: probabilistic Earley

\[
\begin{align*}
\text{ROOT} & \rightarrow \text{NP} \circ \\
1 & \quad 1 \\
\text{NP} & \rightarrow \text{NP} \circ \text{PP} \\
\frac{2}{3} \times \frac{2}{2} & \quad \frac{2}{3} \\
\text{NP} & \rightarrow \text{NP} \circ \text{PP} \\
\frac{1}{3} \times \frac{3}{2} & \quad \frac{1}{3} \\
\text{ROOT} & \rightarrow \text{NP} \\
1 & \quad 1 \\
\text{NP} & \rightarrow \text{Det} \circ \text{N} \\
\frac{2}{3} \times \frac{2}{3} & \quad \frac{2}{3} \\
\text{NP} & \rightarrow \text{Det} \circ \text{N} \\
\frac{2}{3} \times \frac{2}{3} & \quad \frac{2}{3} \\
\text{P} & \rightarrow \text{near} \\
\frac{2}{9} & \quad 1 \\
\text{PP} & \rightarrow \text{P} \circ \text{NP} \\
\frac{2}{9} & \quad 1 \\
\text{Det} & \rightarrow \text{the} \\
1 & \quad 1 \\
\text{N} & \rightarrow \text{dog} \\
\frac{2}{3} \times \frac{2}{3} & \quad \frac{2}{3} \\
\text{N} & \rightarrow \text{dog} \\
\frac{2}{3} & \quad \frac{2}{3} \\
\text{P} & \rightarrow \text{near} \\
\frac{2}{9} & \quad 1 \\
\text{the} & \\
1 & \\
\text{dog} & \\
\frac{2}{3} \times \frac{2}{3} & \quad \frac{2}{3} \\
\text{near} & \\
\frac{2}{9} & \quad 1 \\
\text{the} & \\
1 & 
\end{align*}
\]
Efficient incremental parsing: probabilistic Earley
Efficient incremental parsing: probabilistic Earley

<table>
<thead>
<tr>
<th></th>
<th>Det → the</th>
<th>NP → Det N</th>
<th>Det → the</th>
<th>NP → Det N</th>
<th>P → near</th>
<th>NP → Det N</th>
<th>PP → P NP</th>
<th>NP → PP PP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1/3 × 1/3</td>
<td>2/9 × 1/3</td>
<td>2/9 × 1/3</td>
<td>2/9 × 1/3</td>
<td>2/9 × 1/3</td>
<td>2/9 × 1/3</td>
<td>2/9 × 1/3</td>
<td>2/9 × 1/3</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2/3</td>
<td>1</td>
<td>2/3</td>
<td>1</td>
<td>2/3</td>
<td>1</td>
<td>2/3</td>
</tr>
</tbody>
</table>

The diagram shows the probabilistic Earley parser for the sentence "the dog near the".
Efficient incremental parsing: probabilistic Earley

\[
\begin{align*}
\text{ROOT} & \rightarrow \text{NP}_o \\
\frac{4}{9} & \quad \frac{4}{9} \\
\text{Det} & \rightarrow \circ \text{the} \\
1 & \quad 1 \\
\text{NP} & \rightarrow \text{NP}_o \circ \text{PP} \\
\frac{2}{9} & \quad \frac{4}{27} \\
\text{NP} & \rightarrow \circ \text{Det} \circ \text{N} \\
\frac{2}{3} & \times \frac{3}{2} & \frac{2}{3} \\
\text{NP} & \rightarrow \circ \text{NP} \circ \text{PP} \\
\frac{1}{3} & \times \frac{3}{2} & \frac{1}{3} \\
\text{ROOT} & \rightarrow \circ \text{NP} \\
\frac{1}{3} & \quad \frac{1}{3} \\
\text{N} & \rightarrow \circ \text{cat} \\
\frac{2}{9} & \quad 1 \\
\text{N} & \rightarrow \circ \text{dog} \\
\frac{2}{3} & \quad \frac{2}{3} \\
\text{NP} & \rightarrow \text{Det}_o \circ \text{N} \\
1 & \quad \frac{2}{3} \\
\text{Det} & \rightarrow \circ \text{the} \\
1 & \quad 1 \\
\text{P} & \rightarrow \circ \text{near} \\
\frac{2}{9} & \quad 1 \\
\text{PP} & \rightarrow \circ \text{P} \circ \text{NP} \\
\frac{2}{9} & \times \frac{1}{3} & \times \frac{3}{2} & \frac{1}{3} \\
\text{NP} & \rightarrow \circ \text{NP} \circ \text{PP} \\
\frac{2}{9} & \times \frac{2}{3} & \times \frac{3}{2} & \frac{2}{3} \\
\text{NP} & \rightarrow \text{Det}_o \circ \text{N} \\
\frac{2}{9} & \quad \frac{2}{3} \\
\text{Det} & \rightarrow \circ \text{the} \\
\frac{2}{9} & \quad 1 \\
\text{P} & \rightarrow \circ \text{near} \\
\frac{2}{9} & \quad 1 \\
\text{Det} & \rightarrow \circ \text{the} \\
\frac{2}{9} & \quad 1
\end{align*}
\]
If you have just processed word $w_i$, then the prefix probability of $w_1...i$ can be obtained by summing all forward probabilities of items that have the form $X \rightarrow \alpha \circ w_i \beta$.
If you have just processed word $w_i$, then the prefix probability of $w_1...i$ can be obtained by summing all forward probabilities of items that have the form $X \rightarrow \alpha \circ w_i \beta$

In our example, we see:

\[
\begin{align*}
P(\text{the}) &= 1 \\
P(\text{the dog}) &= \frac{2}{3} \\
P(\text{the dog near}) &= \frac{2}{9} \\
P(\text{the dog near the}) &= \frac{2}{9}
\end{align*}
\]
If you have just processed word $w_i$, then the prefix probability of $w_1...i$ can be obtained by summing all forward probabilities of items that have the form $X \rightarrow \alpha \circ w_i \beta$

In our example, we see:

\[
P(\text{the}) = 1 \\
P(\text{the dog}) = \frac{2}{3} \\
P(\text{the dog near}) = \frac{2}{9} \\
P(\text{the dog near the}) = \frac{2}{9}
\]

Taking the ratios of these prefix probabilities can give us conditional word probabilities.
Probabilistic Earley as an “eager” algorithm

- From the *inside probabilities* of the states on the chart, the posterior distribution on (incremental) trees can be directly calculated.
- This posterior distribution is *precisely* the correct result of the application of Bayes’ rule.
- Hence, probabilistic Earley is also performing rational disambiguation.
- Hale (2001) called this the “eager” property of an incremental parsing algorithm.
Probabilistic Earley algorithm: key ideas

- We want to use probabilistic grammars for both disambiguation and calculating probability distributions over upcoming events.
- Infinitely many trees can be constructed in polynomial time and space.
- The prefix probability of the string is calculated in the process.
- By taking the log-ratio of two prefix probabilities, the surprisal of a word in its context can be calculated.
Probabilistic Earley algorithm: key ideas

- We want to use probabilistic grammars for both disambiguation and calculating probability distributions over upcoming events.
- Infinitely many trees can be constructed in polynomial time ($O(n^3)$) and space ($\text{O}(\cdot)$).
- The **prefix probability** of the string is calculated in the process.
- By taking the log-ratio of two prefix probabilities, the surprisal of a word in its context can be calculated.
Probabilistic Earley algorithm: key ideas

- We want to use probabilistic grammars for both disambiguation and calculating probability distributions over upcoming events.
- Infinitely many trees can be constructed in polynomial time ($O(n^3)$) and space ($O(n^2)$).
- The prefix probability of the string is calculated in the process.
- By taking the log-ratio of two prefix probabilities, the surprisal of a word in its context can be calculated.
You can read about the (non-probabilistic) Earley algorithm in (Jurafsky and Martin, 2000, Chapter 13)

Prefix probabilities can also be calculated with an extension of the CKY algorithm due to Jelinek and Lafferty (1991)
References I


Putting the *psycho* in *psycholinguistics*

How can we use these models to study human language processing?
Syntactic disambiguation

• Consider: *The women discussed the dogs on the beach*
  • Is the discussion on the beach?
Human forced-choice preference (Ford et al., 1982)

The women discussed the dogs on the beach.

The women kept the dogs on the beach.

90%

95%
Jurafsky, 1996

- A product of experts probabilistic parser
  - Model 1: a PCFG
  - Model 2: a verb valency model (use your corpus to count how often each verb appears with each possible argument structure)
- \( P(\text{sentence}) = P_{\text{PCFG}}(\text{sentence})P_{\text{valence}}(\text{sentence}) \)
- Note: if one were doing this today, then there are better options (e.g., lexicalized PCFG)
Human forced-choice preference (Ford et al., 1982)

Probabilistic parser: PCFG + verb valency (Jurafsky, 1996)

The women discussed the dogs on the beach.

The women kept the dogs on the beach.
Local ambiguity/“Garden paths”

- Consider: *The horse raced past the barn fell.*
  - (versus: *The car driven past the barn crashed.*)
- By the end of the sentence, this is actually unambiguous... but people still have problems!
- Jurafsky suggests: incremental parsing via *beam search* – only keep the top $n$ parses
  - (see also Crocker & Brants, 2000)
- “Garden path” sentences are ones where the correct parse was *pruned*
- Prediction: Garden path processing difficulty will occur at disambiguating region
- ...But how do we check?
A methodological digression

• The ideal data for studying language comprehension should be:
  • Incremental
  • Naturalistic
  • Unconscious

• How about *per-word reading time*?
Reading time: technicalities

Two main types:

- *Self-paced reading* (moving window style)
Reading time: technicalities

Two main types:

- Self-paced reading (moving window style)
Reading time: technicalities

Two main types:

- *Self-paced reading (moving window style)*
Reading time: technicalities

Two main types:

- Self-paced reading (moving window style)
Reading time: technicalities

Two main types:

- *Self-paced reading (moving window style)*

---
cat
---
Reading time: technicalities

Two main types:

- *Self-paced reading (moving window style)*
Reading time: technicalities

Two main types:

- *Self-paced reading (moving window style)*
Reading time: technicalities

Two main types:

- *Self-paced reading (moving window style)*
Reading time: technicalities

Two main types:

- *Self-paced reading (moving window style)*
Reading time: technicalities

Two main types:

- *Self-paced reading (moving window style)*
Reading time: technicalities

Two main types:

- **Self-paced reading (moving window style)**
  - Cheap and easy to do
  - Easy to interpret (more or less)
  - Still not the most natural situation

- **Eye-tracking**
  - More naturalistic
Eye-tracking: complications

- We want to know how long it takes the brain to process each word
- We can measure where the eyes are pointing
- These are not the same...
Eye movements

• Eyes each move in 3 dimensions (up/down, left/right, rotational)
• At least 5 distinct systems for controlling them
• BUT, only one is really relevant to reading: Saccadic eye movements
Saccadic eye movement

- Alternation between:
  - *Fixation*: eye is still and gathering sensory information (~100-500 ms)
  - *Saccade*: extremely fast “jump” to a new position (~20-50 ms), little or no sensory information available
Why saccadic eye movement?

- There is only a small part of our visual field where we can see details – the **fovea**
Why saccadic eye movement?

- There is only a small part of our visual field where we can see details – the fovea.
Eye-tracking for reading

The cat sat on the mat.

Regression (~15-25% of fixations)

- What do the eyes tell us about the brain?
  - Just and Carpenter, 1980: *The eye-mind hypothesis*: what you're looking at is what you're processing
  - Seems to work pretty well in practice
The cat sat on the mat.

- How long did it take to read *on*?
- How long did it take to read *the*?
  - First fixation time: ✗
  - First pass reading time: ✗ + ✗

Reading time: technicalities

Two main types:

- **Self-paced reading** *(moving window style)*
  - Cheap and easy to do
  - Easy to interpret (more or less)
  - Still not the most natural situation

- **Eye-tracking**
  - More naturalistic
Two main types:

- **Self-paced reading (moving window style)**
  - Cheap and easy to do
  - Easy to interpret (more or less)
  - Still not the most natural situation

- **Eye-tracking**
  - More naturalistic
  - Therefore, more complicated...
Reading time: technicalities

Two main types:

- **Self-paced reading (moving window style)**
  - Cheap and easy to do
  - Easy to interpret (more or less)
  - Still not the most natural situation

- **Eye-tracking**
  - More naturalistic
  - Therefore, more complicated...
  - But, we can work around that in practice
Reading time: technicalities

Two main types:

- **Self-paced reading (moving window style)**
  - Cheap and easy to do
  - Easy to interpret (more or less)
  - Still not the most natural situation

- **Eye-tracking**
  - More naturalistic
  - Therefore, more complicated...
  - But, we can work around that in practice

- Either way: we get per-word reading times
Back to garden path sentences!

- Prediction of beam search theory: reading time should increase at the disambiguating region.

\[\text{The horse raced past the barn fell.}\]

Frazier and Rayner (1982) tested sentences like:

\[\text{Wherever Alice walks her dog men follow. Wherever Alice walks her dog will follow.}\]
Back to garden path sentences!

Wherever Alice walks, her dog will follow. Men follow her dog, will follow.

Not significant

Highly significant

(Frazier & Rayner, 1982)
More subtle ambiguities

- The cop arrested... Ambiguity makes this slow
  - by the detective was guilty of taking bribes.
- The crook arrested...
  - by the detective was guilty of taking bribes.

Ambiguity does not affect speed here

(McRae et al., 1998)
More subtle ambiguities

- Probably not a pruning effect
- Narayanan & Jurafsky (2002): an attention shift explanation: incur a reading time cost whenever new data causes a switch in the most-probable interpretation
- Use Bayes net model to compute probabilities:
More subtle ambiguities

This causes a flip from MV to RR

- *The cop arrested...*
  - ..*by the detective* was guilty of taking bribes.
- *The crook arrested...*
  - ..*by the detective* was guilty of taking bribes.

RR was always preferred; no flip

(McRae et al., 1998; Narayanan & Jurafsky, 2002)
So is probability just for ambiguity?

*My brother came inside to*

- In general, predictable words are read faster than unpredictable words.
- A good mathematical model:
  \[
  \text{Surprisal}(w_i) = \log \frac{1}{P(w_i|w_1...i-1, \text{Context})}
  \]

(Hale, 2001)
Surprisal: example

Sentences from Konieczny (2000):

1. **Er** hat **den Abgeordneten** begleitet, und …
   He has the delegate escorted, and …
   “He escorted the delegate, and …”
2. **Er** hat **den Abgeordneten** ans **Rednerpult** begleitet, und …
   He has the delegate to the lectern escorted, and …
   “He escorted the delegate to the lectern, and …”
3. **Er** hat **den Abgeordneten** an das **große** Rednerpult begleitet, und …
   He has the delegate to the big lectern escorted, and …
   “He escorted the delegate to the large lectern, and …”

- Memory theories: understanding the **verb** requires recalling the **subject** and other **arguments** => these keep getting harder and harder!
  

- Surprisal: the more you've seen, the more you're expecting a verb => these get easier and easier!
  
  (Levy, 2008)
Surprisal example

Memory theories don't make precise numerical predictions, but something like this...

(Levy, 2008)
Reading time ~ Surprisal

\[ \text{Surprisal}(w_i) = \log \frac{1}{P(w_i|w_1...i-1, \text{Context})} \]

- This seems oddly specific... How do we know?
  - Dundee eye-tracking corpus: publically available, 10 participants × 50,000 words
  - ~425,000 total fixations
  - ~200,000 first fixations
  - Estimate word probabilities with Kneser-Ney trigram model

(Smith & Levy, 2008)
Reading time $\sim$ Surprisal

Individual participants

Reading time is linearly proportional to surprisal

unique contribution of probability to first fixation time individually by subject plus 95% confidence intervals (bootstrapped)

(Smith & Levy, 2008)
Reading time ~ Surprisal

• But why?
  • Relative entropy?
Surprisal as relative entropy

Relative entropy (= Kullback-Leibler divergence): a fundamental information-theoretic measure of the distance between two generative probability distributions

\[ D(P_i || P_{i-1}) = \sum_i P_i(I) \log \frac{P_{i-1}(I)}{P_i(I)} \]

Intuitively, the penalty paid by encoding one distribution with a different one

It turns out that relative entropy over interpretation distributions before and after \( wi = \log \frac{1}{P_{i-1}(w_i)} \) (surprisal!)

Surprisal can thus be thought of as cost of shifting probability mass

Compare to attention shift/reranking cost in ambiguity resolution

(Levy, 2008)
Reading time ~ Surprisal

• But why?
  • Relative entropy?
Reading time ~ Surprisal

• But why?
  • Relative entropy?
  • Optimal sensory discrimination?
Surprisal as optimal discrimination

The children went outside to
My brother came inside to

\[ P(\text{word} | \text{context}) \times P(\text{word} | \text{visual input}) = \text{Expected} \]

Less expected words need minimal visual evidence to reach a certainty threshold, which is slow

Expected words need more visual evidence to reach a certainty threshold, which is fast

(Norris, 2006)
Reading time ~ Surprisal

• But why?
  • Relative entropy?
  • Optimal sensory discrimination?
Reading time ~ Surprisal

• But why?
  • Relative entropy?
  • Optimal sensory discrimination?
  • Optimal preparation?
Surprisal as optimal preparation

come together

(Smith & Levy, 2008)
Surprisal as optimal preparation

(Smith & Levy, 2008)
Surprisal as optimal preparation

- Only resources spent preparing for *together* end up being useful
- Optimal trade-off: spend more resources preparing for words with higher probability
- Scale invariance produces log relationship

(Smith & Levy, 2008)
Reading time ~ Surprisal

- But why?
  - Relative entropy?
  - Optimal sensory discrimination?
  - Optimal preparation?
  - *(your theory here)*

Is that the only puzzle? *No...*
Memory vs. surprisal: *round 2*

- *The reporter who sent* the photographer to...
- *The reporter who the photographer sent* to...

**Surprisal predicts this will be slow** (because until here, you could have been in the other sentence)

**Memory predicts this will be slow** (because we saw *the reporter* a long time ago)

...and it's right!

(Grodner & Gibson, 2005; Levy, 2008)
...Memory and surprisal?

- Demberg & Keller (2008)
  - Used a dependency parser to automatically parse the Dundee corpus
  - Then used these dependencies to estimate memory/integration costs, and estimated surprisal
  - Found:
    - Memory effects on verbs (mostly)
    - AND
    - Surprisal effects generally
Conclusion 1

- Probabilistic models explain some otherwise mysterious phenomena...
- ...but it's not the end of the story!

Conclusion 2

- Solving these problems will require coffee
Memory constraints: a theoretical puzzle

• # Logically possible analyses grows at best exponentially in sentence length

• Exact probabilistic inference with context-free grammars can be done efficiently in $O(n^3)$

• But…
  • Requires probabilistic locality, limiting conditioning context
  • Human parsing is linear—that is, $O(n)$—anyway

• So we must be restricting attention to some subset of analyses

• **Puzzle**: how to choose and manage this subset?
  • Previous efforts: $k$-best beam search (Crocker & Brants, 2000; Roark, 2001, 2004)
  • Here, we’ll explore the particle filter as a model of limited-parallel approximate inference

*Levy, Reali, & Griffiths, 2009, NIPS*
Sequential Monte Carlo for incremental observations

Let $x_i$ be observed data, $z_i$ be unobserved states

- For parsing: $x_i$ are words, $z_i$ are incremental structures

Suppose that after $n-1$ observations we have the distribution over interpretations $P(z_{n-1}|x_1...n-1)$

After next observation $x_n$, represent the next distribution $P(z_n|x_1...n)$ inductively:

$$P(z_n| x_1...n) \propto P(x_n|z_n) \sum_{z_{n-1}} P(z_n|z_{n-1}) P(z_{n-1}|x_1...n-1)$$

- Sample $z_n$ from $P(z_n|z_{n-1})$, and reweight by $P(x_n|z_n)$
Particle filter with probabilistic grammars

S → NP VP 1.0
NP → N 0.8
NP → N RRC 0.2
RRC → Part N 1.0
VP → V N 1.0
N → women 0.7
N → sandwiches 0.3
V → brought 0.4
V → broke 0.3
V → tripped 0.3
Part → brought 0.1
Part → broken 0.7
Part → tripped 0.2
Adv → quickly 1.0
Resampling in the particle filter

- With the naïve particle filter, inferences are highly dependent on initial choices
  - Most particles wind up with small weights
  - Region of dense posterior poorly explored
- Especially bad for parsing
  - Space of possible parses grows (at best) exponentially with input length
Resampling in the particle filter

- With the naïve particle filter, inferences are highly dependent on initial choices
  - Most particles wind up with small weights
  - Region of dense posterior poorly explored
- Especially bad for parsing
  - Space of possible parses grows (at best) exponentially with input length
- We handle this by resampling at each input word
Simple garden-path sentences

The woman brought the sandwich from the kitchen tripped

MAIN VERB (it was the woman who brought the sandwich)

REDUCED RELATIVE (the woman was brought the sandwich)

- Posterior initially misled away from ultimately correct interpretation
- With finite # of particles, recovery is not always successful
Solving a puzzle

A-S  Tom heard the gossip wasn’t true.
A-L  Tom heard the gossip about the neighbors wasn’t true.
U-S  Tom heard that the gossip wasn’t true.
U-L  Tom heard that the gossip about the neighbors wasn’t true.

Previous empirical finding: ambiguity induces difficulty…
…but so does the length of the ambiguous region

Our linking hypothesis:
Proportion of parse failures at the disambiguating region should increase with sentence difficulty

Frazier & Rayner, 1982; Tabor & Hutchins, 2004
Another example (Tabor & Hutchins 2004)

As the author wrote the essay the book grew.
As the author wrote the book grew.
As the author wrote the essay the book describing Babylon grew.
As the author wrote the book describing Babylon grew.
Resampling-induced drift

- In ambiguous region, observed words aren’t strongly informative ($P(x_i|z_i)$ similar across different $z_i$)
- But due to resampling, $P(z_i|x_i)$ will drift
- One of the interpretations may be lost
- The longer the ambiguous region, the more likely this is
Ambiguity matters…

But the length of the ambiguous region also matters!
Human results (offline rating study)

![Graphs showing the proportion of parse successes as a function of the number of particles for NP/S and NP/Z.]

- **NP/S**
  - U-S
  - A-S
  - U-L
  - A-L

- **NP/Z**
  - U-S
  - A-S
  - U-L
  - A-L

**Mean difficulty rating**
- U-S
- A-S
- U-L
- A-L
Approximate-inference summary

- Approximate inference—either beam-search or particle-filter—is a way of giving approximate probabilistic processing in linear time.
- Loss of analyses accounts better for garden-pathing than sheer surprisal does.
- Stochasticity in the particle filter could account for “digging-in” effects.
Bounds on rationality?

- Thus far, we have painted a pretty picture of humans of rational comprehenders.
- Lots of information sources can be usefully brought to bear in the difficult task of language comprehension.
- People *rationally* and *incrementally* use *all the information available*.
- We have lots of evidence that people do this often.

> “Put the apple on the towel in the box.” (Tanenhaus et al., 1995)

- But now we’ll challenge this with a new puzzle.
Anatomy of *ye olde garden path sentence*

- Classic example of incrementality in comprehension

```
"Main Verb"

S

NP

VP

S

"Reduced Relative"

VP

"The horse raced past the barn fell."

(The criminal arrested by the detective confessed.)

- People fail to understand it most of the time
- People are likely to *misunderstand* it—e.g.,
  - “What’s a *barn fell*?”
  - The horse *that* raced past the barn fell
  - The horse raced past the barn *and* fell
A puzzle for incremental inference

• Try to understand this sentence:

(a) The coach smiled at the player tossed the frisbee.

...and contrast this with:

(b) The coach smiled at the player thrown the frisbee.

(c) The coach smiled at the player who was thrown the frisbee.

(d) The coach smiled at the player who was tossed the frisbee.

• Readers boggle at “tossed” in (a), but not in (b-d)

(Tabor et al., 2004)
Why is *tossed/thrown* interesting?

- As with classic garden-paths, part-of-speech ambiguity leads to misinterpretation
  - *The horse raced past the barn…fell*
    - verb?
    - participle?
- But now context “should” rule out the garden path:
  - *The coach smiled at the player tossed…*
    - verb?
    - participle?
- A substantial challenge for rational, incremental models: *failure to condition on relevant context*
Meeting this challenge

- These effects of merely “locally coherent” chunks within a sentence have attracted considerable attention.
- Tabor et al., (2004); Tabor & Hutchins (2004) treat these effects within an essentially dynamical-systems tree-substitution grammar approach.
- We’ll now discuss two recent models of our own that account for these effects in different ways…
- …plus new empirical research deriving from the modeling work.
Weighted finite-state automata and intersection with PCFGs

Roger Levy

UC San Diego
Department of Linguistics

1 June 2010
Weighted finite-state automata

A WEIGHTED FINITE-STATE AUTOMATON (wFSA) is a tuple $(Q, V, q_0, q_f, T, W)$ such that:

- $Q$ is a finite set of states;
- $V$ is a finite set of input symbols, including the empty string $\epsilon$;
- $q_0 \in Q$ is the start state;
- $q_f \in Q$ is the end state;
- $T$ is a set of transitions of the form $x : q \rightarrow q'$ where $x \in V$ and $q, q' \in Q$;
- $W$ is a mapping from $T$ to weights.

(There are alternative formulations with multiple final states, starting-state costs, and/or final-state costs, but they all have the same basic properties as what we show here.)
The weights in a wFSA can be drawn from any of a number of **semirings** (Mohri, 1997; Goodman, 1999).

A semiring is a set over which two binary operators are defined: $\oplus$ and $\otimes$, intuitively behaving more or less like arithmetic $+$ and $\times$.

Here, we’ll consider weights drawn from:

- The non-negative real (or negative-log) semirings, corresponding to (negative-log) probabilities; or
- The tropical semiring, corresponding to highest-probability derivations.

<table>
<thead>
<tr>
<th>Semiring</th>
<th>$x \oplus y$</th>
<th>$x \otimes y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real</td>
<td>$x + y$</td>
<td>$x \times y$</td>
</tr>
<tr>
<td>Negative-log</td>
<td>$- \log(e^{-x} + e^{-y})$</td>
<td>$x + y$</td>
</tr>
<tr>
<td>Tropical</td>
<td>$\max(x, y)$</td>
<td>$x + y$</td>
</tr>
</tbody>
</table>
A path through a wFSA is a sequence of transitions drawn from $T$ taking you from the start state $q_0$ to the final state $q_f$.

The weight of a path through a wFSA is the $\oplus$-sum of the weights of all transitions on the path.

The weight $s(w)$ of a string $w$ is the sum of the weights of all the paths that generate the string.

Note that this can be more than one (or even infinitely many) paths even if the wFSA is deterministic, due to $\epsilon$ transition.
For a real or negative-log semiring wFSA, the total weight of all accepted strings is called its PARTITION FUNCTION $Z$

$$Z = \sum_w s(w)$$

If $Z$ is finite, then the PROBABILITY of a string $w$ in a wFSA is its weight normalized by the partition function:

$$P(w) = \frac{s(w)}{Z}$$

If the normalizing constant is 1 (that is, $\otimes$ identity), then the wFSA can be said to be a proper PROBABILISTIC wFSA.

Stronger condition: the wFSA is STATEWISE PROPER if for every non-final state, sum of outgoing path weights is 1 (final state may alternatively have no outgoing paths; see also Smith and Johnson, 2007)
Weighted finite-state automata V

- Of course, in general there are too many possible paths to calculate the partition function $Z$ exactly by enumeration.
- However, there are dynamic programs for computing $Z$ efficiently:
  - Generic single-source algorithm (Mohri, 2002; exponential-time)
  - Generalized all-pair shortest distance (Lehmann, 1977; $O(n^3)$ in principle, sometimes worse in practice; Allauzen, p.c.)
- One can also use fixed-point or other methods to approximate $Z$ (Stolcke, 1995; Nederhof and Satta, 2008)
Intersection with PCFGs

- A widely known property of FSAs is that both FSAs and CFGs are **CLOSED UNDER INTERSECTION** with them (Bar-Hillel et al., 1964)
- That is, for any two FSAs $a_1, a_2$, there is some FSA that accepts only those strings which are accepted by both $a_1$ and $a_2$
- Likewise, for any CFG $g$ and any FSA $a$ and, there is some CFG that accepts only those strings which are accepted by both $a$ and $g$
- Furthermore, there are constructive procedures for creating the intersection FSA/CFG, and the paths/rules bear a close resemblance to the original rules
Additionally, closure under intersection extends to the *weighted* regime

For any two wFSAs $a_1, a_2$, there is a wFSA $a_{12}$ such that for all strings $w$, $s_{a_{12}}(w) = s_{a_1}(w) \otimes s_{a_2}(w)$
Additionally, closure under intersection extends to the \textit{weighted} regime.

For any two wFSAs $a_1, a_2$, there is a wFSA $a_{12}$ such that for all strings $w$, $s_{a_{12}}(w) = s_{a_1}(w) \otimes s_{a_2}(w)$.

Now consider a PCFG $g$ and a wFSA $a$ in the real semiring. There is some \textit{weighted} CFG $g \cap$ such that for all strings $w$, $s_{g \cap}(w) = P_g(w) \times s_{a}(w)$.
Intersection with PCFGs II

- Additionally, closure under intersection extends to the *weighted* regime.
- For any two wFSAs $a_1, a_2$, there is a wFSA $a_{12}$ such that for all strings $w$, $s_{a_{12}}(w) = s_{a_1}(w) \otimes s_{a_2}(w)$.
- Now consider a PCFG $g$ and a wFSA $a$ in the real semiring. There is some *weighted* CFG $g \cap$ such that for all strings $w$, $s_{g \cap}(w) = P_g(w) \times s_a(w)$.
- (The same would hold if $g$ and $a$ had probabilities and weights in the negative-log semiring respectively.)
Intersection with PCFGs II

- Additionally, closure under intersection extends to the weighted regime
- For any two wFSAs $a_1, a_2$, there is a wFSA $a_{12}$ such that for all strings $w$, $s_{a_{12}}(w) = s_{a_1}(w) \otimes s_{a_2}(w)$
- Now consider a PCFG $g$ and a wFSA $a$ in the real semiring. There is some weighted CFG $g \cap$ such that for all strings $w$, $s_{g \cap}(w) = P_g(w) \times s_a(w)$
- (The same would hold if $g$ and $a$ had probabilities and weights in the negative-log semiring respectively.)
- Furthermore, if $Z$ is finite for each constituent grammar, then $Z$ is finite in the intersection grammar (n.b. for most empirically estimated PCFGs, $Z = 1$; Chi and Geman, 1998)
Intersection with PCFGs II

- Additionally, closure under intersection extends to the *weighted* regime
- For any two wFSAs $a_1, a_2$, there is a wFSA $a_{12}$ such that for all strings $w$, $s_{a_{12}}(w) = s_{a_1}(w) \otimes s_{a_2}(w)$
- Now consider a PCFG $g$ and a wFSA $a$ in the real semiring. There is some *weighted* CFG $g_\cap$ such that for all strings $w$, $s_{g_\cap}(w) = P_g(w) \times s_a(w)$
  (The same would hold if $g$ and $a$ had probabilities and weights in the negative-log semiring respectively.)
- Furthermore, if $Z$ is finite for each constituent grammar, then $Z$ is finite in the intersection grammar (n.b. for most empirically estimated PCFGs, $Z = 1$; Chi and Geman, 1998)
- If the $Z$'s are finite, then $g_{ab}$ and $g_\cap$ are thus *probabilistic* grammars (Abney et al., 1999; Chi, 1999; Smith and Johnson, 2007)
Construction of the weighted CFG/FSA intersection

- **“PCFG phrase” rules:** For each rule $X \rightarrow Y^1 \ldots Y^n$ with probability $p$ in the PCFG, and each state sequence $r_0, \ldots, r_n$ in the wFSA, put a rule

$$r_0 X r_n \rightarrow_{r_0} Y^1 r_1 \ldots r_{n-1} Y^n r_n$$

into $g_\cap$ with weight $p$
Construction of the weighted CFG/FSA intersection

- **“PCFG phrase” rules:** For each rule $X \rightarrow Y^1 \ldots Y^n$ with probability $p$ in the PCFG, and each state sequence $r_0, \ldots, r_n$ in the wFSA, put a rule

  $$r_0 X r_n \rightarrow r_0 Y^1 r_1 \ldots r_{n-1} Y^n r_n$$

  into $g \cap$ with weight $p$

- **“wFSA bottoming-out” rules:** For each transition $x : r \rightarrow r'$ in the wFSA with weight $s$, put the rule

  $$r X r' \rightarrow x$$

  into $g \cap$ with weight $s$
Construction of the weighted CFG/FSA intersection

- **“PCFG phrase” rules**: For each rule \( X \rightarrow Y^1 \ldots Y^n \) with probability \( p \) in the PCFG, and each state sequence \( r_0, \ldots, r_n \) in the wFSA, put a rule

\[
r_0X_{rn} \rightarrow r_0 Y^1 r_1 \cdots r_{n-1} Y^n r_n
\]

into \( g \cap \) with weight \( p \)

- **“wFSA bottoming-out” rules**: For each transition \( x : r \rightarrow r' \) in the wFSA with weight \( s \), put the rule

\[
rX_{r'} \rightarrow x
\]

into \( g \cap \) with weight \( s \)

- You’re done!
Example of PCFG/wFSA intersection

- Imagine you’ve heard the incomplete sentence

  \textit{dogs static cats fought}...
Example of PCFG/wFSA intersection

- Imagine you’ve heard the incomplete sentence
  \[ \text{dogs static cats fought} \ldots \]

- Let’s consider one set of possibilities based purely on input:
  \[ \text{dogs and cats fought} \ldots \]
  \[ \text{dogs and dogs and cats fought} \ldots \]
  \[ \text{dogs and cats and cats fought} \ldots \]
  \[ \ldots \]
Example of PCFG/wFSA intersection

- Imagine you’ve heard the incomplete sentence
  \[ \text{dogs static cats fought} \ldots \]

- Let’s consider one set of possibilities based purely on input:
  \[ \text{dogs and cats fought} \ldots \]
  \[ \text{dogs and dogs and cats fought} \ldots \]
  \[ \text{dogs and cats and cats fought} \ldots \]

- We can represent these possibilities succinctly with a wFSA:
Example of PCFG/wFSA intersection III

How do we interpret this partial input???
Example of PCFG/wFSA intersection III

How do we interpret this partial input???

Grammar-based interpretation: a simple PCFG

1  ROOT → S
1  S → NP VP
1  VP → fought

0.3  NP → cats
0.3  NP → dogs
0.4  NP → NP Conj NP
1   Conj → and
Example of PCFG/wFSA intersection III

How do we interpret this partial input???
Grammar-based interpretation: a simple PCFG

Grammar-based interpretation of partial sentences is just (weighted) intersection (Lang, 1988, 1991; Hale, 2006; Levy, 2008)
How do we interpret this partial input???

Grammar-based interpretation: a simple PCFG

```
1  ROOT  → S
1  S      → NP  VP
1  VP     → fought
```

Grammar-based interpretation of partial sentences is just (weighted) intersection (Lang, 1988, 1991; Hale, 2006; Levy, 2008)

We can use a bottom-up agenda-based parser (Shieber et al., 1995) to determine what rules will actually matter in the grammar
Example of PCFG/wFSA intersection III

ROOT → S
S → NP VP
NP → NP Conj NP

ROOT → S
S → NP VP
NP → NP Conj NP
Conj → and
NP → cats
NP → dogs

NP → dogs
0
dogs

ROOT → S
S → NP VP
NP → NP Conj NP
Conj → and
NP → cats
NP → dogs

NP → cats
1
cats

VP → fought
2
fought

ROOT → S
S → NP VP
NP → NP Conj NP
Conj → and
NP → cats
NP → dogs

NP → dogs
3
After pruning away the “useless” rules (those that bottom out in zero weight), we get the following intersection weighted CFG:

1 _ \text{ROOT}_3 \rightarrow_0 S_3
1 _ S_3 \rightarrow_0 NP_1 \_ VP_3
1 _ VP_3 \rightarrow_1 fought_3
0.3 _ NP_1 \rightarrow_0 dogs_1
0.3 _ NP_2 \rightarrow_1 cats_2
0.4 _ NP_1 \rightarrow_0 NP_1 \_ Conj_1 \_ NP_1
1 _ Conj_1 \rightarrow_1 and_1
0.3 _ NP_1 \rightarrow_1 dogs_1
0.3 _ NP_1 \rightarrow_1 cats_1
0.4 _ NP_1 \rightarrow_1 NP_1 \_ Conj_1 \_ NP_1

This grammar is improper, but by computing the partition function of each category we can turn it into a PCFG (Chi, 1999; Nederhof and Satta, 2008)
Local coherences under rational comprehension

Comprehension could still be rational if we relax some of our assumptions today: two models that get local coherences in a rational framework.

- Model 1: relax assumption that all contextual information is immediately available
- Model 2: relax assumption that input is free of noise

Plan
- present each model
- experiments showing how they get local coherence effects
Local coherences under rational comprehension

Comprehension could still be rational

- if we relax some of our assumptions
- today: two models that get local coherences in a rational framework
Local coherences under rational comprehension

Comprehension could still be rational

- if we relax some of our assumptions
- today: two models that get local coherences in a rational framework
- Model 1: relax assumption that all contextual information is immediately available
Local coherences under rational comprehension

Comprehension could still be rational

- if we relax some of our assumptions
- today: two models that get local coherences in a rational framework
- Model 1: relax assumption that all contextual information is immediately available
- Model 2: relax assumption that input is free of noise
Local coherences under rational comprehension

Comprehension could still be rational

- if we relax some of our assumptions
- today: two models that get local coherences in a rational framework
- Model 1: relax assumption that all contextual information is immediately available
- Model 2: relax assumption that input is free of noise

Plan

- present each model
- experiments showing how they get local coherence effects
Model 1 (Bicknell & Levy, 2009, NAACL)

Goals

▶ propose a new model of local coherence effects under rational sentence comprehension
  ▶ bottom-up information is rapidly available
  ▶ but comprehenders ultimately arrive at normative probabilities (combining bottom-up and top-down information)

▶ show that an implemented version makes the correct predictions for the Tabor et al. data
The intuition: New sentence built of familiar parts

The coach smiled at the player tossed a frisbee.
The intuition: New sentence built of familiar parts

D

The coach smiled at the player tossed a frisbee
The intuition: New sentence built of familiar parts

N

The coach smiled at the player tossed a frisbee
The intuition: New sentence built of familiar parts

The coach smiled at the player tossed a frisbee
The intuition: New sentence built of familiar parts

The coach smiled at the player tossed a frisbee
The intuition: New sentence built of familiar parts

The coach smiled at the player tossed a frisbee
The intuition: New sentence built of familiar parts

The coach smiled at the player tossed a frisbee.
The coach smiled at the player tossed a frisbee
The intuition: New sentence built of familiar parts

The coach smiled at the player tossed a frisbee

Diagram:

```
VP
  / \ NP
   \  
    VBD D N
```
The intuition: New sentence built of familiar parts

The coach smiled at the player tossed a frisbee
The intuition: New sentence built of familiar parts

The coach smiled at the player tossed a frisbee
The intuition: Parsing as belief update

Notice: These likely structures can be incompatible

```
PP
  
NP
  P D N
  | | |
at the player

S
  
NP VP
  D N VBD
  | | |
the player tossed
```
The intuition: Parsing as belief update

Notice: These likely structures can be incompatible
The intuition: Parsing as belief update

Notice: These likely structures can be incompatible

what to throw away?
The intuition: Parsing as belief update

Notice: These likely structures can be incompatible

what to throw away?

this is belief update
The intuition: Parsing as belief update

The belief update process

▶ integrating a new word or words $w_i^j$ into a parse is belief update
▶ start: precomputed prior beliefs about structure that $w_i^j$ will have
▶ end: posterior beliefs about structure $w_i^j$ has in context
▶ hypothesis: difficulty increases with the size of the belief update
The model formally

for each syntactic category $X$:

- begin with ‘bottom-up’ prior that $w^j_i$ forms the beginning of it
  - Prior: $P(X_{i}^{k \geq j} | w^j_i)$
The model formally

for each syntactic category $X$:

- begin with ‘bottom-up’ prior that $w_i^j$ forms the beginning of it
  - Prior: $P(X_i^{k \geq j} | w_i^j)$

- integrate with ‘top-down’ knowledge to reach a posterior probability of $X$ spanning $w_i^j$ given all words seen
  - Posterior: $P(X_i^{k \geq j} | w_0^j)$
The model formally

for each syntactic category $X$:

- begin with ‘bottom-up’ prior that $w_i^j$ forms the beginning of it
  - Prior: $P(X_i^{k\geq j}|w_i^j)$

- integrate with ‘top-down’ knowledge to reach a posterior probability of $X$ spanning $w_i^j$ given all words seen
  - Posterior: $P(X_i^{k\geq j}|w_0^j)$

- $M_{ij} \equiv$ the amount of modification required to update prior to posterior when integrating $w_i^j$
An intuitive example

The coach smiled at the player tossed...

Artificial probabilities for illustrative purposes

\[ \begin{bmatrix} S & .91 \\ NP & .22 \\ \vdots \end{bmatrix} \]

the player tossed
An intuitive example

The coach smiled at the player tossed...
An intuitive example

The coach smiled at the player tossed...

Artificial probabilities for illustrative purposes

\[
\begin{bmatrix}
S & .91 \\
NP & .22
\end{bmatrix}
\]

Prior

the player tossed

The coach smiled at the player tossed

\[
\begin{bmatrix}
S & .01 \\
NP & .95
\end{bmatrix}
\]

Posterior

\(M_{ij}\) high
An intuitive example

The coach smiled at the player thrown . . .

\[
\begin{bmatrix}
S & 0.3 \\
NP & 0.9 \\
\vdots & \\
\end{bmatrix}
\]

Prior

the player thrown

Artificial probabilities for illustrative purposes
An intuitive example

The coach smiled at the player thrown . . .

Artificial probabilities for illustrative purposes

\[
\begin{bmatrix}
S & .3 \\
NP & .9 \\
\vdots & \\
\end{bmatrix}
\]

Prior

\[
\begin{bmatrix}
S & .01 \\
NP & .95 \\
\vdots & \\
\end{bmatrix}
\]

Posterior

The coach smiled at the player thrown
An intuitive example

The coach smiled at the player thrown . . .

Artificial probabilities for illustrative purposes

\[
\begin{bmatrix}
S & .3 \\
NP & .9 \\
\vdots & \\
\end{bmatrix}
\]

Prior

\[
\begin{bmatrix}
S & .01 \\
NP & .95 \\
\vdots & \\
\end{bmatrix}
\]

Posterior

\( M_{ij} \) low
Model 1

Implementation

- use probabilistic context-free grammar (PCFG)
- calculate prior and posterior for each nonterminal category
  - neither may look obvious, but we can reduce them to forward probabilities
- measure $M_{ij}$ as Kullback-Leibler (K-L) divergence from the prior to the posterior, summed over categories

$$M_{ij} \overset{\text{def}}{=} \sum_{X \in N} D\left( \begin{array}{l} \text{Posterior} \\ P(X_i^{k\geq j} \mid w_0^i) \end{array} \middle\| \begin{array}{l} \text{Prior} \\ P(X_i^{k\geq j} \mid w_i^j) \end{array} \right)$$
Experiment

See whether this intuition is borne out among Tabor et al.’s items

Methods

- defined a small PCFG
- estimating probabilities from parsed Brown corpus
- used 12 of Tabor et al.’s 20 items (so model knew verb & syntax)
- same 4 conditions as Tabor et al.:
  - Lexical ambiguity: *tossed vs. thrown*
  - Relative clause reduction: *who was vs. ∅*
- used an equal combination of 1-, 2-, and 3-word sizes of $w_i^j$
- prediction: $M_{ij}$ will be highest for ∅ *tossed*
Experiment: Results

Summed K-L Divergence (bits)

- △ - tossed
- ○ - who was tossed
- ▲ - thrown
- ⋯ - who was thrown

at the player tossed/thrown a frisbee
Experiment: Results

Summed K-L Divergence (bits)

- △ - tossed
- ○ - who was tossed
- ▲ - thrown
- • - who was thrown

at
the
player
tossed/
thrown

Residual Reading Time (ms)

- △ - tossed (A/R)
- △ - who was tossed (A/U)
- ■ - thrown (U/R)
- ■ - who was thrown (U/U)
Model 2: Uncertainty about input

- Model 1 relaxed assumption that all contextual information is immediately available
- Model 2 instead relaxes assumption that input is perfectly formed and free of noise
- Plan:
  1. Motivate and describe this model
  2. Show how it predicts local coherence effects
Model 2: Uncertainty about input

- State of the art models for ambiguity resolution ≈ probabilistic incremental parsing
- Simplifying assumption:
  - Input is *clean* and *perfectly-formed*
  - No uncertainty about input is admitted
- Intuitively seems patently wrong…
  - We sometimes *misread* things
  - We can also *proofread*
- Leads to two questions:
  1. What might a model of sentence comprehension under uncertain input look like?
  2. What interesting consequences might such a model have?
Today: a first-cut answer

1. What might a model of sentence comprehension under uncertain input look like?
2. What interesting consequences might such a model have?

- **First**: a simple noisy-channel model of rational sentence comprehension under uncertain input
- **Then**: show how it can predict local coherence effects
- We use probabilistic context-free grammars (PCFGs) and weighted finite-state automata (WFSAs) to instantiate the model
The noisy-channel model

• Say we use a weighted generative grammar $G$ to parse a sentence $w$. We get a posterior over structures $T$:

$$P_G(T|w) = \frac{P(T, w)}{P(w)} \propto P(T, w)$$

• If we don’t observe a sentence but only a noisy input $I$:

$$P_G(T|I) \propto \sum_w P(I|T, w) P(w|T) P(T)$$

• Posterior over possible sentences:

$$P_G(w|I) \propto \sum_T P(I|T, w) P(w|T) P(T)$$

Levy, 2008 (EMNLP)
The noisy-channel model (II)

- This much is familiar from the parsing of speech (Hall & Johnson, 2003, 2004; Johnson & Charniak, 2004)
- Alternative scenario: we know the true sentence \( w^* \) but not observed input \( I \) (e.g., the study of reading)
- Expected inferences of the comprehender marginalize over the input \( I \):

\[
P(w | w^*) = \int_I P_C(w | I, w^*) P_T(I | w^*) \, dI
\]

\[\propto Q(w, w^*)\]
Representing noisy input

• How can we represent the type of noisy input generated by a word sequence?

• *Probabilistic finite-state automata* (pFSAs; Mohri, 1997) *are a good model*

\[
\text{vocab} = a, b, c, d, e, f
\]

```
Input symbol    Log-probability (surprisal)
```

“Word 1 is a or b, and I have no info about Word 2”
A generative probabilistic grammar determines beliefs about which strings are likely to be seen:

- Probabilistic Context-Free Grammars (PCFGs; Booth, 1969)
- Probabilistic Minimalist Grammars (Hale, 2006)
- Probabilistic Finite-State Grammars (Mohri, 1997; Crocker & Brants 2000)

In position 1, \{a,b,c,d\} equally likely; but in position 2:

- \{a,b\} are usually followed by e, occasionally by f
- \{c,d\} are usually followed by f, occasionally by e
Combining grammar & uncertain input

- Bayes’ Rule says that the evidence and the prior should be combined (multiplied)
- For probabilistic grammars, this combination is the formal operation of weighted intersection

Grammar affects beliefs about the future
Revising beliefs about the past

• When we’re uncertain about the future, grammar + partial input can affect beliefs about what will happen.
• With uncertainty of the past, grammar + future input can affect beliefs about *what has already happened*.
The noisy-channel model (FINAL)

\[ P(w|w^*) \propto P_C(w)Q(w, w^*) \]

- For \( Q(w, w^*) \): a WFSA based on Levenshtein distance between words (\( K_{LD} \)):

\[
\text{Cost}(a \text{ cat sat}) = 0 \\
\text{Cost}(\text{sat a sat cat}) = 8 \quad \text{Result of } K_{LD} \text{ applied to } w^* = a \text{ cat sat}
\]
Near-neighbors make the “incorrect” analysis “correct”:

- Any of these changes makes \textit{tossed} a main verb!!!

- The coach smiled at the player \textbf{tossed} the frisbee

Hypothesis: the boggle at “tossed” involves \textit{what the comprehender wonders whether she might have seen}
The core of the intuition

- Grammar & input come together to determine two possible “paths” through the partial sentence: 

  • tossed is more likely to happen along the bottom path
  • This creates a large shift in belief in the tossed condition
  • thrown is very unlikely to happen along the bottom path
  • As a result, there is no corresponding shift in belief
Ingredients for the model

\[ P(w | w^*) \propto P_C(w) Q(w, w^*) \]

- \( Q(w, w^*) \) comes from \( K_{LD} \) (with minor changes)
- \( P_C(w) \) comes from a probabilistic grammar
- We need one more ingredient:
  - a **quantified signal** of the alarm induced by word \( w_i \) about changes in beliefs about the past
Quantifying alarm about the past

- **Relative Entropy** (KL-divergence) is a natural metric of change in a probability distrib. (Levy, 2008; Itti & Baldi, 2005)

- Our distribution of interest is *probabilities over the previous words in the sentence*

- Call this distribution $P_i(w_{[0,j]})$

- The change induced by $w_i$ is the *error identification signal* $EIS_i$, defined as

$$D \left( P_i \left( w_{[0,i]} \right) \right) \| \left( P_{i-1} \left( w_{[0,i]} \right) \right)$$

  - new distribution
  - old distribution
Error identification signal: example

- Measuring change in beliefs about the past:

$$\sum_{w_1} P_2(w_1) \log \frac{P_2(w_1)}{P_1(w_1)} = \begin{cases} w_1 = b & 0.28 (1 - 1.82) + 0.72 (1 - 0.48) = 0.14 \\ w_1 = c & \end{cases}$$
Results on local-coherence sentences

- Locally coherent: *The coach smiled at the player tossed the frisbee*
- Locally incoherent: *The coach smiled at the player thrown the frisbee*

(All sentences of Tabor et al. 2004 with lexical coverage in model)
Prediction 2: neighborhood manipulation

- Novel prediction: changing the neighborhood of the context can change the EIS

The coach smiled at the player tossed the frisbee

The coach smiled toward the player tossed the frisbee

- Substituting toward for at should reduce the EIS
- In free reading, we should see less tendency to regress from tossed when the EIS is small

(Levy, Bicknell, Slattery & Rayner, 2009)
Experimental design

In a free-reading eye-tracking study, we crossed *at*/toward with *tossed*/thrown:

The coach smiled *at* the player *tossed* the frisbee
The coach smiled *at* the player *thrown* the frisbee
The coach smiled *toward* the player *tossed* the frisbee
The coach smiled *toward* the player *thrown* the frisbee

Prediction: interaction between preposition & ambiguity in some subset of:

- Early-measure RTs at critical region *tossed*/thrown
- First-pass regressions out of critical region
- Go-past time for critical region
- Regressions into *at*/toward
Model predictions

(The coach smiled at/toward the player tossed/thrown the frisbee)
Experimental results

The coach smiled at the player tossed...
Summary & Conclusions

• First model dealing with input uncertainty & implications for sentence comprehension

• **Empirical support** from an eye-tracking experiment
  • First evidence of fully bidirectional word↔syntax inference

• Noise and perceptual uncertainty in the system aren’t just afterthoughts
  – They introduce fundamentally new kinds of inferences!
What’s missing

- How and why does EIS change eye movements?
- Ideal: make EIS-like behavior fall out of a rational model of reading
- Here: propose a rational framework for eye movements in reading
  - cf. popular non-rational computational models of eye movements in reading: *E-Z Reader* (Reichle, Rayner, & Pollatsek, 2003) and SWIFT (Engbert et al., 2005)
A new framework for eye movements in reading

Readers have a diverse range of goals:

▶ understanding the point of the passage
▶ going as fast as possible while getting the comprehension question

These goals are all accomplished by getting info about the text's identity

Identifying the text

▶ two sources of information
  ▶ language knowledge (prior)
  ▶ visual input (likelihood)

Central hypothesis: eyes move to obtain visual information about text, which helps identify it and thus achieve reader goals
A new framework for eye movements in reading

Readers have a diverse range of goals:

- understanding the point of the passage
- going as fast as possible while getting the comprehension question

These goals are all accomplished by getting info about the text's identity:

Identifying the text

- two sources of information
  - language knowledge (prior)
  - visual input (likelihood)

The normative way to combine these: Bayesian inference

Central hypothesis: eyes move to obtain visual information about text, which helps identify it and thus achieve reader goals.
A new framework for eye movements in reading

Readers have a diverse range of goals:

▶ understanding the point of the passage
A new framework for eye movements in reading

Readers have a diverse range of goals:

- understanding the point of the passage
- going as fast as possible while getting the comprehension question
A new framework for eye movements in reading

Readers have a diverse range of goals:

- understanding the point of the passage
- going as fast as possible while getting the comprehension question

These goals are all accomplished by getting info about the text's identity.
A new framework for eye movements in reading

Readers have a diverse range of goals:

- understanding the point of the passage
- going as fast as possible while getting the comprehension question

These goals are all accomplished by getting info about the text's identity

Identifying the text

- two sources of information
  - language knowledge
  - visual input
A new framework for eye movements in reading

Readers have a diverse range of goals:

- understanding the point of the passage
- going as fast as possible while getting the comprehension question

These goals are all accomplished by getting info about the text's identity

Identifying the text

- two sources of information
  - language knowledge (prior)
  - visual input (likelihood)
- normative way to combine these: Bayesian inference
A new framework for eye movements in reading

Readers have a diverse range of goals:

- understanding the point of the passage
- going as fast as possible while getting the comprehension question

These goals are all accomplished by getting info about the text's identity

Identifying the text

- two sources of information
  - language knowledge (prior)
  - visual input (likelihood)
- normative way to combine these: Bayesian inference
- prior is known, so eyes move for visual input
A new framework for eye movements in reading

Readers have a diverse range of goals:

▶ understanding the point of the passage
▶ going as fast as possible while getting the comprehension question

These goals are all accomplished by getting info about the text's identity

Identifying the text

▶ two sources of information
  ▶ language knowledge (prior)
  ▶ visual input (likelihood)
▶ normative way to combine these: Bayesian inference
▶ prior is known, so eyes move for visual input

Central hypothesis: eyes move to obtain visual information about text, which helps identify it and thus achieve reader goals
1. present an implemented model of reading within this framework
2. report a simulations with it, demonstrating how model can be flexibly adapted to different reader goals
An implemented model

Framework is general

- start with prior expectations for the text (language knowledge)
- move eyes to get visual input
- update beliefs about text identity as visual input arrives
An implemented model

Framework is general

- start with prior expectations for the text (language knowledge)
- move eyes to get visual input
- update beliefs about text identity as visual input arrives

Pieces of a model in this framework
An implemented model

Framework is general

- start with prior expectations for the text (language knowledge)
- move eyes to get visual input
- update beliefs about text identity as visual input arrives

Pieces of a model in this framework

1. formal problem of reading: possible actions a reader can take
An implemented model

Framework is general

- start with prior expectations for the text (language knowledge)
- move eyes to get visual input
- update beliefs about text identity as visual input arrives

Pieces of a model in this framework

1. formal problem of reading: possible actions a reader can take
2. visual input
An implemented model

Framework is general

- start with prior expectations for the text (language knowledge)
- move eyes to get visual input
- update beliefs about text identity as visual input arrives

Pieces of a model in this framework

1. formal problem of reading: possible actions a reader can take
2. visual input
3. behavior policy: how the model decides between the actions
An implemented model

Framework is general

- start with prior expectations for the text (language knowledge)
- move eyes to get visual input
- update beliefs about text identity as visual input arrives

Pieces of a model in this framework

1. formal problem of reading: possible actions a reader can take
2. visual input
3. behavior policy: how the model decides between the actions

Next up: describing each of these pieces for our current model
The model: Formal reading problem
The model: Formal reading problem

Assume discrete timesteps and on each:

- get visual input about sentence around the fixated character
- update beliefs about sentence identity (vision + context)
- choose an action
The model: Formal reading problem II

Four possible actions

- continue fixating current location
- initiate a forward saccade to position $t$
- initiate a backward saccade to position $t$
- stop reading the sentence
The model: Formal reading problem II

Four possible actions

- continue fixating current location *(deterministic)*
- initiate a forward saccade to position $t$
- initiate a backward saccade to position $t$
- stop reading the sentence *(deterministic)*
The model: Formal reading problem II

Four possible actions

- continue fixating current location (deterministic)
- initiate a forward saccade to position $t$
- initiate a backward saccade to position $t$
- stop reading the sentence (deterministic)

When a saccade is initiated

- one timestep of delay (saccade execution lags behind initiation)
- then, landing position normally distributed around $t$
  - (variance increases with intended distance)
The model: Visual input
The model: Visual input

- acuity curve given by asymmetric Gaussian (as in SWIFT)
- narrow foveal input (14 highest chars)
- wide peripheral input (20 highest chars)
The model: Visual input

Foveal input

- veridical information about word boundaries
- noisy information about letter identity (noise related to acuity)
- no letter confusability (following Norris, 2006)

Peripheral input

- veridical information about word boundaries only
Implementation

Noisy visual input

- represent characters as 26-d vectors (each: one 1, 25 0s)
- visual input $\sim$ Gaussian
  - mean = true identity
  - diagonal covariance matrix (magnitude relates to visual acuity)

Beliefs as wFSTs

- grammar (prior) represented with wFST
- visual input (likelihood) given by wFST
  - each arc gives liklihood of character identity at that position
- posterior = normalized intersection of grammar and visual input
Behavior policy: How to choose actions
Behavior policy: How to choose actions

- model’s belief state is a distribution over possible sentences
- decisions made based on *parameterized* policy sensitive to current belief state
Behavior policy: How to choose actions

- model’s belief state is a distribution over possible sentences
- decisions made based on *parameterized* policy sensitive to current belief state
- policy might be sensitive to many things
  - expectations about upcoming material
  - ambiguity in structural analysis
  - semantics
  - relevance to expected comprehension question
  - ...

Today's simple policy

- confidence in a character position: probability of most likely character under model beliefs
- move left to right, bringing up confidence in each position to $\alpha$
- make regression if confidence about a previous position falls below $\beta$
  (if $\beta = 0$, never make a regression)
Behavior policy: How to choose actions

- model’s belief state is a distribution over possible sentences
- decisions made based on parameterized policy sensitive to current belief state
- policy might be sensitive to many things
  - expectations about upcoming material
  - ambiguity in structural analysis
  - semantics
  - relevance to expected comprehension question
  - ...

Today’s simple policy

- CONFIDENCE in a character position: probability of most likely character under model beliefs
- move left to right, bringing up confidence in each position to $\alpha$
- make regression if confidence about a previous position falls below $\beta$
  - (if $\beta = 0$, never make a regression)
Experiment

Motivation

▶ one advantage of this framework: actions are goal-directed
▶ thus, we can ask a new question:
  ▶ How should reading behavior change depending on the reader’s goal?
Experiment

Motivation

- one advantage of this framework: actions are goal-directed
- thus, we can ask a new question:
  - How should reading behavior change depending on the reader’s goal?

Specifically, using simple policy family and simple goal functions

- simple policy family: how should $\alpha$ and $\beta$ change?
- goal functions: relative values of timesteps $T$ vs. accuracy $L$
Experiment

Motivation

▶ one advantage of this framework: actions are goal-directed
▶ thus, we can ask a new question:
  ▶ How should reading behavior change depending on the reader’s goal?

Specifically, using simple policy family and simple goal functions

▶ simple policy family: how should $\alpha$ and $\beta$ change?
▶ goal functions: relative values of timesteps $T$ vs. accuracy $L$

Experiment: find optimal values of $\alpha$ and $\beta$ for three goal functions
Experiment: Methods

Task

- simulate reading on sentences from the Schilling et al. (1998) corpus
- measure average timesteps $T$ and accuracy $L$ (log prob.)
Experiment: Methods

Task
- simulate reading on sentences from the Schilling et al. (1998) corpus
- measure average timesteps $T$ and accuracy $L$ (log prob.)

Model
- bigram language model trained on British National Corpus
- implemented with weighted finite state automata using OpenFST (Allauzen et al., 2007)
Experiment: Methods II

Formal goal functions

- consider linear combinations of $L$ and $T$: $L(1 - \gamma) - T\gamma$
  where $\gamma \in [0, 1]$ gives weighting of time
- we optimize for 3 goals: $\gamma \in \{.025, .1, .4\}$
Experiment: Methods II

Formal goal functions

- consider linear combinations of $L$ and $T$: $L(1 - \gamma) - T\gamma$
  where $\gamma \in [0, 1]$ gives weighting of time
- we optimize for 3 goals: $\gamma \in \{.025, .1, .4\}$

Optimization

- use the **PEGASUS** method (Ng & Jordan, 2000) to estimate performance for a given $[\alpha, \beta]$
- (essentially, just average performance over sentences from Schilling)
- then, we use standard hillclimbing techniques to find optimum
Experiment: Results

<table>
<thead>
<tr>
<th>$\gamma$</th>
<th>$\alpha$</th>
<th>$\beta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>.025</td>
<td>.90</td>
<td>.99</td>
</tr>
<tr>
<td>.1</td>
<td>.36</td>
<td>.80</td>
</tr>
<tr>
<td>.4</td>
<td>.18</td>
<td>.38</td>
</tr>
</tbody>
</table>

Optimization worked

- $\alpha$ and $\beta$ decrease as time valued more
Experiment: Results

<table>
<thead>
<tr>
<th>$\gamma$</th>
<th>$\alpha$</th>
<th>$\beta$</th>
<th>Timesteps</th>
<th>Accuracy (Prob.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>.025</td>
<td>.90</td>
<td>.99</td>
<td>41.2</td>
<td>-0.02 ($p \approx .98$)</td>
</tr>
<tr>
<td>.1</td>
<td>.36</td>
<td>.80</td>
<td>25.8</td>
<td>-0.90 ($p \approx .41$)</td>
</tr>
<tr>
<td>.4</td>
<td>.18</td>
<td>.38</td>
<td>16.4</td>
<td>-4.59 ($p \approx .01$)</td>
</tr>
</tbody>
</table>

Optimization worked

- $\alpha$ and $\beta$ decrease as time valued more
- mean $T$ and $L$ decrease as time valued more
Conclusion

A new framework for eye movements in reading

- readers achieve diverse goals by getting info about text identity
- text identity is normatively given by Bayesian inference combining
  - language knowledge (prior)
  - visual input (likelihood)
Conclusion

A new framework for eye movements in reading

- readers achieve diverse goals by getting info about text identity
- text identity is normatively given by Bayesian inference combining
  - language knowledge (prior)
  - visual input (likelihood)
- hypothesis: eye movements are produced to get visual input
- see also
Future directions

Technical directions

- so far, very simple two-parameter behavioral policies
- investigating richer, more realistic classes of policies
- investigating using richer language models
Future directions

Technical directions

- so far, very simple two-parameter behavioral policies
- investigating richer, more realistic classes of policies
- investigating using richer language models

Modeling reader goals

- Exp. demonstrated model’s flexibility to adapt to different reader goals
- allows asking new questions:
Future directions

Technical directions

- so far, very simple two-parameter behavioral policies
- investigating richer, more realistic classes of policies
- investigating using richer language models

Modeling reader goals

- Exp. demonstrated model’s flexibility to adapt to different reader goals
- allows asking new questions: How should reading behavior change as
  - …accuracy is valued more or less relative to time?
Future directions

Technical directions

- so far, very simple two-parameter behavioral policies
- investigating richer, more realistic classes of policies
- investigating using richer language models

Modeling reader goals

- Exp. demonstrated model’s flexibility to adapt to different reader goals
- allows asking new questions: How should reading behavior change as
  - ...accuracy is valued more or less relative to time?
  - ...readers learn comprehension questions all ask about direct object?
Future directions

Technical directions

- so far, very simple two-parameter behavioral policies
- investigating richer, more realistic classes of policies
- investigating using richer language models

Modeling reader goals

- Exp. demonstrated model’s flexibility to adapt to different reader goals
- allows asking new questions: How should reading behavior change as...
  - ...accuracy is valued more or less relative to time?
  - ...readers learn comprehension questions all ask about direct object?
  - ...readers learn that the text uses a lot of difficult words?
Future directions

Technical directions

- so far, very simple two-parameter behavioral policies
- investigating richer, more realistic classes of policies
- investigating using richer language models

Modeling reader goals

- Exp. demonstrated model’s flexibility to adapt to different reader goals
- allows asking new questions: How should reading behavior change as
  - ...accuracy is valued more or less relative to time?
  - ...readers learn comprehension questions all ask about direct object?
  - ...readers learn that the text uses a lot of difficult words?
  - ...a number of other exciting questions!
Uncertain input and surprisal

- Surprisal theory: the reading time for a word is a function of its probability
- And obviously, its probability is a function of its identity...
- ...But we don't know its identity! Then what?
- Maybe the brain makes an educated guess

This section: (Smith & Levy, 2010)
Average neighborhood surprisal

Best guess: $S = 2 \times 0.9 + 5 \times 0.1 = 2.3$
Average neighborhood surprisal

Best guess: $S = 2 \times 0.1 + 5 \times 0.9 = 4.7$
Model: visual noise

- Simplify: assume number of letters is known; only their identity is in doubt
- One free parameter scales the confusability matrix to set overall level of noise
- Use Bayes' rule to combine top-down prior from context with bottom-up visual information
For letters:

\[ P(\text{input} \mid \text{letter}) = \frac{P(\text{letter} \mid \text{input})P(\text{input})}{P(\text{letter})} \]

From (rescaled) confusion matrix

Assumed uniform

For words:

\[ P(\text{word}_i \mid \text{input, context}) = \frac{P(\text{word}_i \mid \text{context})P(\text{input} \mid \text{word}_i, \text{context})}{P(\text{input} \mid \text{context})} \]

\[ = \frac{P(\text{word}_i \mid \text{context}) \prod_j P(\text{input}_j \mid \text{word}_{ij})}{Z} \]

Conditional independence

Top-down prior from context

Letter model from above
Average neighborhood surprisal

- Defined as: the average of the surprisal of all words in the language in the current context, weighted by the probability that each word is, in fact, the currently viewed word

- Can define average neighborhood $X$ for any word property – frequency, imageability, ...

- Prediction: average neighborhood surprisal (ANS) is a better predictor of reading times than raw surprisal (RS)
Results: First fixations

• Best-fitting noise level was rather low: ~66% accuracy at naming single letters

• ANS and RS highly correlated: \( R^2 = 0.96 \)

• Nevertheless, **ANS unambiguously pre-empts RS** in regression model:
  - ANS: \( t(182155) = -4.2, p < 0.001 \)
  - RS: \( t(182155) = -0.49, \text{n.s.} \)

• Prediction: **confirmed**
Second fixations

• Surprisal effect is also seen in the duration of re-fixations. We predict that these should show a reduced effect of visual noise relative to first fixations

• Result 1: ANS still beats RS:
  – ANS: $t(42010) = -4.2, p < 0.001$
  – RS: $t(42010) = 1.8, p = 0.065$

• Result 2: noise parameter is $\sim$half that for first fixations
The next mystery...

- Can define average *neighborhood* $X$ for any word property – frequency, imageability, …
The next mystery...

- Can define *average neighborhood X* for any word property – *frequency*, imageability, …
  - We tried this; it didn't work.
- Conclusion?: surprisal-sensitive brain process is earlier than frequency-sensitive brain process, has less visual data available
- Conclusion?: surprisal-sensitive process is more complicated, must get started sooner
- Conclusion?: visual noise is minimal, frequency effect is part of visual processing, and surprisal noise is really internal representational noise
- ...?
Remaining challenges and future directions

We hope to have convinced you of a few things:

- The picture of human language comprehension as rational, incremental deployment of probabilistic knowledge accounts for many empirical phenomena
- This picture spans both comprehension and production
- Formal tools from computational linguistics are incredibly useful in both theory development and data analysis
- Fine-grained modeling of cognitive and environmental constraints in language processing is important
- There are many open issues ripe for exploration
- Now I’ll take a moment to mention a few outstanding challenges and directions for future work
Models of memory recall

• You may remember this challenge for surprisal:
  - Subject-extracted relative clause
    The reporter who sent the photographer to...
  - Object-extracted relative clause
    The reporter who the photographer sent to...

• Recent work on similarity-based interference seems to identify confusability of preceding NPs as a culprit
  - Object-extracted RCs aren’t always hard (Warren et al., 2002)
    The reporter who I sent to...
  - Direct manipulations show influence of similarity (Gordon et al., 2002)
    Memory list: (table, sink, truck)
    It was the boat that the guy who lived by the sea fixed/sailed...
Integrating memory & expectations

- Lewis & Vasishth (2005) have an incremental (left-corner), serial CFG production system implemented in ACT-R
- New input triggers potentially inaccurate recall of previous material
Demberg & Keller (2009) have an incremental Tree-Adjoining Grammar model with both prediction & verification components.

Retrospective verification cost gives correct predictions for English RCs.
Modeling online pragmatic inference

- Let's go back to the classic visual world experiment

“Put the apple on the towel in the box.” (Tanenhaus et al., 1995)

- What knowledge is required to correctly draw the inference between the context and the correct syntax?
  - Grammatical knowledge
  - Linking linguistic constituents to referents in the visual field
  - **Pragmatics:** when would a speaker use a postmodifier?
Discourse structure

• Most work analyzing syntactic-level phenomena and down has focused on isolated-sentence phenomena
• But that’s not how language is used naturalistically
• Inter-clause and inter-sentence meaning relations are essential
• This has been well-known in psycholinguistics for a while

A burglar broke into a bank carrying some dynamite. He planned to blow up a safe. Once inside he saw that there was a safe with a new lock and a safe/strongbox with an old lock.

The burglar blew up the safe with the…

(Altmann & Steedman, 1988)

• But how to model these effects is open territory
Broad-coverage assessment of models

- Traditional psycholinguistic experiments have tended to use isolated sentences with tightly controlled content & structure
- **Advantages:** your world is simpler
- **Disadvantages:** your world may be mostly imaginary
- Increasing trend toward implementing psycholinguistic models with broad coverage and testing on naturalistic datasets
- You’ve seen this already in many cases
  - Demberg & Keller, 2008; Smith & Levy, 2008
  - Roark et al., 2009
- Knowledge of computational linguistics is essential
Increasing connections between fields

- Computational linguistics and psycholinguistics have had many points of contact throughout the years
- We seem to be on an upswing
  - CUNY 2008: special session on computational models
  - ACL 2010: there’s a psycholinguistics track!
- Help us continue this trend!