Computational Cognitive Science
Lecture 14: Syntactic Surprisal

Chris Lucas
(Slides adapted from Frank Keller’s)

School of Informatics
University of Edinburgh
clucas2@inf.ed.ac.uk

9 November 2017
1 Background
   - Expectations in Sentence Processing
   - Beyond Ambiguity

2 The Surprisal Model
   - Computing Surprisal
   - Results

Reading: Hale (2001).
Garden Paths and Odds

Last time, we saw the model of human parsing proposed by Jurafsky (1996):

- all syntactic trees for a sentence are computed in parallel and assigned probabilities;
- the probabilities are used to rank the set of trees; low probability trees are pruned (no longer considered);
- the set of trees (and their probabilities) is updated incrementally (word by word) as the input comes in;
- if the tree that turns out to be ultimately correct has been pruned, then a garden path occurs.
Garden Paths and Odds

Example garden path sentence:

(1) The horse raced past the barn fell.

**First parse tree:**

\[ P(\text{race, } \langle \text{agent} \rangle) = 0.92 \]

\[
\begin{array}{c}
S \\
\downarrow \\
NP \quad VP \\
\downarrow \quad \downarrow \\
\text{the horse} \quad \text{raced}
\end{array}
\]

\[ P(t_1) = 0.92 \text{ (preferred)} \]
Second parse tree:

\[ P(\text{race, } \langle \text{agent, theme} \rangle) = 0.08 \]

\[ \text{NP} \rightarrow \text{NP XP} \quad 0.14 \]

\[ P(t_2) = 0.0112 \text{ (grossly dispreferred)} \]
In the Jurafsky model, processing difficulty is caused by the ratio of the correct parse to the best parse. For example in (1):

\[ \frac{P(t_1)}{P(t_2)} = \frac{0.92}{0.0112} = 82 : 1 \]

Intuitively, a high ratio means that the parser has a strong expectation about the correct structure.

Maybe this expectation is based not only on two trees (most probable one and correct one) but on all trees of the sentence.

*Syntactic surprisal:* processing difficulty (including garden paths) occurs when the probability distribution over parse trees changes.
There is evidence that expectation plays a role in sentence processing. For instance, when the parser sees an either, it expects an or (Staub & Clifton, 2006):

(2) Peter read either a book or an essay in the school magazine.
(3) Peter read a book or an essay in the school magazine.

The region an essay is read faster in (2) than in (3).

The parser is surprised to see an or if it doesn’t expect it, i.e., if there is no either. Surprisal leads to processing difficulty.
Expectations in Sentence Processing

Intuitively, this is also what is going on in garden paths:

(4)  
   a. The horse raced past the barn fell.
   b. The bird found in the room died.

In (4-a), the parser is surprised when it gets *fell*, as it expected the sentence to end at *barn*. In (4-b), the surprisal at *died* is lower.

(5)  
   a. The complex houses married and single students.
   b. The warehouse fires a dozen employees each year.

In (5-a), the parser is surprised when it gets *married*, as it expected a verb. In (5-b) it assumes *fires* is a verb, and is not surprised.
Beyond Ambiguity

Ambiguity resolution (and garden paths) is not the only thing we want to model in sentence processing.

Some sentences cause difficulty even though not ambiguous:

(6)  
   a. The reporter that attacked the senator admitted the error.  
   b. The reporter that the senator attacked admitted the error.

The object relative clause (ORC) in (6-b) is more difficult to process than the subject relative clause (SRC) in (6-a).

To be modeled: reading time differences on the relative clause verb and noun phrase (Staub, 2010).
Beyond Ambiguity

Empirical Data (Staub 2010, Expt 1)

Word
Reading Time in ms
0 100 200 300 400 500 600
***
***
***

rel_pron src_vb det noun orc_vb main_vb
SRC ORC

Reading Time in ms

rel_pron src_vb det noun orc_vb main_vb

***
The Surprisal Model

The surprisal model (Hale, 2001) assumes an *incremental, parallel, probabilistic parser*:

- all syntactic trees for a sentence prefix $w_1 \cdots w_k$ are computed at the same time, and assigned probabilities;
- the set of trees is updated as a new word $w_{k+1}$ comes in; trees no longer compatible with the input are removed;
- surprisal measures the change in the probability distribution as trees are removed (disconfirmed) when $w_{k+1}$ is processed;
- if $w_{k+1}$ disconfirms trees with a large probability mass (high surprisal), then processing difficulty occurs.
Surprisal is defined in terms of $P(T|w_1 \cdots w_k)$, the probability distribution over trees $T$ given a sentence prefix $w_1 \cdots w_k$, but comes down to:

$$- \log(P(w_{k+1}|w_k))$$

We’ve already seen surprisal in the Frank et al. model (object individuation).
The Surprisal Model

Levy argues that a good measure of belief change is the Kullback-Leibler divergence (relative entropy) between the syntactic expectations before and after seeing the new word.

The KL divergence between two distributions \( P \) and \( Q \) is:

\[
D(P \| Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)}
\]  

(1)

For a mathematical argument that Surprisal is equivalent to KL divergence, see Levy (2008).
The Surprisal Model

The KL divergence at word $w_{k+1}$ is:

$$\sum_T P(T|w_1 \cdots w_{k+1}) \log \frac{P(T|w_1 \cdots w_{k+1})}{P(T|w_1 \cdots w_k)}$$

(2)

This captures the difference in beliefs before and after seeing $w_{k+1}$. 
Computing Surprisal

One advantage of using surprisal rather than computing KL divergence directly is that Surprisal doesn’t depend on which representation we use; we just need to compute $P(w_{k+1} \mid w_1 \cdots w_k)$. We could use:

- an incremental parser which computes probabilities over trees;
- an $n$-gram model, which computes probabilities over sequences of words;
- intermediate cases, such as a model which computes probabilities over part of speech sequences (a tagger);
- a recurrent neural network.

However, when modeling human language processing, we are interested in the cognitive process that leads to surprisal (e.g., an $n$-gram model doesn’t tell us much about that).
Computing Surprisal

The prefix probability $P(w_1 \cdots w_k)$ can be obtained from a parser by summing over all trees compatible with the prefix:

$$P(w_1 \cdots w_k) = \sum_T P(T, w_1 \cdots w_k)$$ (3)

We can now formulate surprisal in terms of prefix probabilities:

$$S_{k+1} = -\log \frac{P(w_1 \cdots w_{k+1})}{P(w_1 \cdots w_k)} = -\log \frac{\sum_T P(T, w_1 \cdots w_{k+1})}{\sum_T P(T, w_1 \cdots w_k)}$$ (4)

This is how surprisal is computed in practice.
Surprisal: Example

Assume we want to compute the prefix probability for:

(7) The reporter who ...

The prefix probability by definition is:

\[ P(\text{the, reporter, who}) = \sum_T P(T, \text{the, reporter, who}) \]

Assume that there is only one tree. We compute its probability using a PCFG, i.e., by multiplying the probs of the rules in \( T \):

\[ P(T, \text{the, reporter, who}) = \prod_i P(\text{rule}_i) \]
Assume the following syntactic tree:

```
S
  /\    /
 NP  VP
  /\    /
 NP  SBAR
  /\    /
 DT NN  WHNP
 /\    |
 The reporter who
  |
       S
       /\    /
      WP  ...
```
An example for a PCFG that generates this tree:

<table>
<thead>
<tr>
<th>Example</th>
<th>Rule</th>
<th>Rule probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>The reporter who</td>
<td>S → VP NP</td>
<td>( p = 0.6 )</td>
</tr>
<tr>
<td>who ...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>The reporter</td>
<td>NP → NP SBAR</td>
<td>( p = 0.004 )</td>
</tr>
<tr>
<td>who ...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>The reporter</td>
<td>NP → DT NN</td>
<td>( p = 0.5 )</td>
</tr>
<tr>
<td>who</td>
<td>DT → the</td>
<td>( p = 0.7 )</td>
</tr>
<tr>
<td>who ...</td>
<td>NN → reporter</td>
<td>( p = 0.0002 )</td>
</tr>
<tr>
<td>who</td>
<td>SBAR → WHNP S</td>
<td>( p = 0.12 )</td>
</tr>
<tr>
<td>who</td>
<td>WHNP → WP</td>
<td>( p = 0.2 )</td>
</tr>
<tr>
<td>who</td>
<td>WP → who</td>
<td>( p = 0.8 )</td>
</tr>
</tbody>
</table>
Results

To evaluate surprisal, we need an probabilistic parser. The incremental top-down parser of Roark (2001) is often used.

Evaluation procedure:

- train the parser on a training corpus (e.g., Penn Treebank);
- take experimental materials from psycholinguistic experiments;
- parse them using the parser, compute the surprisal values for each sentence;
- compare these to the reading time results for the sentence (typically by-condition averages).
Results: *either ... or*

Compare reading times for *either ... or* sentences against surprisal:

Surprisal successfully models the data.
Results: Relative Clauses

Compare reading times for relative clauses against surprisal values:

Empirical Data (Staub 2010, Expt 1)

Surprisal Predictions

Surprisal successfully models only the difference at the NP.
To model the difference at the verb, we need to add a distance-based memory cost component (Demberg, Keller, & Koller, 2013).
Garden paths still work (Hale, 2001):
Results: Garden Paths

Compare reduced relative clause (garden path) against unreduced relative clause (not a garden path):

Subject Relative Clause

Reduced Relative Clause

Log [previous prefix | current prefix]

The banker told about the buy-back resigned.
Summary

- The human sentence processor builds up expectations about the input;
- if these expectations are not met, surprisal ensues, which manifests itself as processing difficulty;
- mathematically, surprisal is the change in the probability distribution over possible trees from one word to the next;
- it can be computed based on the prefix probabilities returned by a probabilistic parser;
- the surprisal model accounts for garden path sentences, but also for processing difficulty not related to ambiguity.
References


