Computational Psycholinguistics, Day 1
Introduction and a classic computational model of sentence production

Roger Levy

UC San Diego
Department of Linguistics

ESSLLI 2009
20 July 2009
Psycholinguistics deals with the problem of how humans

1. acquire
2. represent
3. comprehend
4. produce
language.
Computational Psycholinguistics

Psycholinguistics deals with the problem of how humans
1. acquire
2. represent
3. comprehend
4. produce
language.

In this class, we will study these problems from a computational perspective.
Psycholinguistics deals with the problem of how humans
1. acquire
2. represent
3. comprehend
4. produce
language.

In this class, we will study these problems from a computational perspective.

(in this class, mostly 3–4).
Class goals

- Overview the literature and major areas in which computational psycholinguistic research is carried out
- Get acquainted with some of the key models and their empirical support
- Gain experience in working out the details of a model from the papers (i.e. read them!)
- Critical analysis of models
What is computational modeling? Why do we do it?

- Any phenomenon involving human behavior is so complex that we cannot hope to formulate a comprehensive theory.
- Instead, we devise a *model* that simplifies the phenomenon to capture some key aspect of it.
What might we use a model for?

Models can serve any of the following (related) functions:
What might we use a model for?

Models can serve any of the following (related) functions:

- **Prediction**: estimating the behavior/properties of a new state/datum on the basis of an existing dataset
What might we use a model for?

Models can serve any of the following (related) functions:

- *Prediction*: estimating the behavior/properties of a new state/datum on the basis of an existing dataset
- *Hypothesis testing*: as a framework for determining whether a given factor has an appreciable influence on some other variable
What might we use a model for?

Models can serve any of the following (related) functions:

- **Prediction**: estimating the behavior/properties of a new state/datum on the basis of an existing dataset
- **Hypothesis testing**: as a framework for determining whether a given factor has an appreciable influence on some other variable
- **Data simulation**: creating artificial data more cheaply and quickly than through empirical data collection
What might we use a model for?

Models can serve any of the following (related) functions:

- **Prediction**: estimating the behavior/properties of a new state/datum on the basis of an existing dataset
- **Hypothesis testing**: as a framework for determining whether a given factor has an appreciable influence on some other variable
- **Data simulation**: creating artificial data more cheaply and quickly than through empirical data collection
- **Summarization**: If phenomenon X is complex but relevant to phenomenon Y, it can be most effective to use a simple model of X when constructing a model of Y
What might we use a model for?

Models can serve any of the following (related) functions:

- **Prediction**: estimating the behavior/properties of a new state/datum on the basis of an existing dataset
- **Hypothesis testing**: as a framework for determining whether a given factor has an appreciable influence on some other variable
- **Data simulation**: creating artificial data more cheaply and quickly than through empirical data collection
- **Summarization**: If phenomenon X is complex but relevant to phenomenon Y, it can be most effective to use a simple model of X when constructing a model of Y
- **Insight**: Most generally, a good model can be explored in ways that give insight into the phenomenon under consideration
Most of the work we will look at involves probabilistic modeling of psycholinguistic phenomena.

Probabilistic effects are pervasive in acquisition, representation, comprehension, and production.

...but we’ll start off with a paper from another tradition: memory in sentence comprehension.
Feedback from you

Please take a moment to fill out a sheet of paper with this info:

- Name (optional)
- School & Program/Department
- Yearstage in program
- Computational Linguistics background
- Psycholinguistics background
- Probability/Statistics background
- Do you know about (weighted) finite-state automata?
- Do you know about (probabilistic) context-free grammars?
- Other courses you’re taking at ESSLLI
- (other side) What do you hope to learn in this class?
A bit of history

The formal revolution in linguistic analysis:

- Symbolic mathematical models for the description of natural language sentences Chomsky (1956, 1957)

Its psychological ramifications:

- Miller (1956): the human working-memory capacity has severe limits on how many items can be stored simultaneously

- Miller and Chomsky (1963): Although human linguistic competence requires transformational grammars, human linguistic performance might require only simpler (even finite-state) grammars
A model and an Hypothesis for Language Structure
Perhaps the very first paper in computational psycholinguistics
Defining the computational problem

There is an infinite number of sentences in any natural language. How is any given sentence produced?
Basic modeling assumptions

- The *grammar* by which a sentence is produced consists of a finite, unordered set of context-free production rules:
  
  \[
  S \rightarrow NP \ VP \\
  NP \rightarrow Det \ N \\
  \ldots
  \]

- An utterance is constructed incrementally—top-down and “left to right”
Yngve’s model has the following components:

- A *computing register* that can store a single symbol
- A *temporary memory* that can store any number of symbols in a stack (first-in, last-out)
- An *output*
The model starts with the root symbol S in the computing register.
The model starts with the root symbol $S$ in the computing register.

One execution cycle consists of the following steps:

1. Copy the symbol in the register to the output.
The model starts with the root symbol S in the computing register.

One execution cycle consists of the following steps:

1. Copy the symbol in the register to the output.
2. If the symbol $X$ in the register is a word,
The model starts with the root symbol S in the computing register.

One execution cycle consists of the following steps:

1. Copy the symbol in the register to the output.
2. If the symbol $X$ in the register is a word,
   2.1 delete it;
The model starts with the root symbol S in the computing register.

One execution cycle consists of the following steps:

1. Copy the symbol in the register to the output.
2. If the symbol X in the register is a word,
   2.1 delete it;
   2.2 if the temporary memory is not empty, move its top symbol to the register;
The model starts with the root symbol S in the computing register.

One execution cycle consists of the following steps:

1. Copy the symbol in the register to the output.
2. If the symbol $X$ in the register is a word,
   2.1 delete it;
   2.2 if the temporary memory is not empty, move its top symbol to the register;
   2.3 otherwise end.
The model in more detail (2)

The model starts with the root symbol S in the computing register. One execution cycle consists of the following steps:

1. Copy the symbol in the register to the output.
2. If the symbol $X$ in the register is a word,
   2.1 delete it;
   2.2 if the temporary memory is not empty, move its top symbol to the register;
   2.3 otherwise end.
3. Otherwise: with the register’s symbol on the left-hand side,
The model starts with the root symbol S in the computing register.

One execution cycle consists of the following steps:

1. Copy the symbol in the register to the output.
2. If the symbol $X$ in the register is a word,
   2.1 delete it;
   2.2 if the temporary memory is not empty, move its top symbol to the register;
   2.3 otherwise end.
3. Otherwise: with the register’s symbol on the left-hand side,
   3.1 choose a production rule $X \rightarrow Y\alpha$;
The model starts with the root symbol $S$ in the computing register.

One execution cycle consists of the following steps:

1. Copy the symbol in the register to the output.
2. If the symbol $X$ in the register is a word,
   2.1 delete it;
   2.2 if the temporary memory is not empty, move its top symbol to the register;
   2.3 otherwise end.
3. Otherwise: with the register’s symbol on the left-hand side,
   3.1 choose a production rule $X \rightarrow Y\alpha$;
   3.2 replace the symbol in the register with $Y$, and put $\alpha$ in the temporary memory.
<table>
<thead>
<tr>
<th>Register</th>
<th>Temporary Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>S</strong></td>
<td></td>
</tr>
</tbody>
</table>

**S → NP VP**
**NP → Det N**
**VP → VP NP**

**Det → the**
**N → dog**
**N → cat**
**V → chased**
<table>
<thead>
<tr>
<th>Register</th>
<th>Temporary Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td></td>
</tr>
</tbody>
</table>

```
S → NP VP
NP → Det N
VP → VP NP

Det → the
N → dog
N → cat
V → chased
```
<table>
<thead>
<tr>
<th>Register</th>
<th>Temporary Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>VP</td>
</tr>
<tr>
<td>NP</td>
<td>VP</td>
</tr>
</tbody>
</table>

*S → NP VP
NP → Det N
VP → VP NP

Det → the
N → dog
N → cat
V → chased
Register

Temporary Memory

S → NP VP
NP → Det N
VP → VP NP

Det → the
N → dog
V → chased

S

NP
<table>
<thead>
<tr>
<th>Register</th>
<th>Temporary Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td></td>
</tr>
<tr>
<td>NP</td>
<td>VP</td>
</tr>
<tr>
<td>Det</td>
<td>N VP</td>
</tr>
</tbody>
</table>

S → NP VP
*NP → Det N
VP → VP NP

Det → the
N → dog
N → cat
V → chased
Register
S
NP
Det

Temporary Memory

S → NP VP
NP → Det N
VP → VP NP

Det → the
N → dog
N → cat
V → chased
```
S → NP VP
NP → Det N
VP → VP NP

*Det → the
N → dog
N → cat
V → chased
```
Register
S
NP
Det
the
N
VP

Temporary Memory
S \rightarrow \text{NP VP}
\text{NP} \rightarrow \text{Det N}
\text{VP} \rightarrow \text{VP NP}

Det \rightarrow \text{the}
N \rightarrow \text{dog}
V \rightarrow \text{chased}

S
  \rightarrow
  \text{NP}
   \rightarrow
     \text{Det N}
       \rightarrow
         \text{I the}
Register

Temporary Memory

S → NP VP
NP → Det N
VP → VP NP

Det → the
*N → dog
N → cat
V → chased
Register

S
NP
Det
the
N
dog

Temporary Memory

S  →  NP  VP
NP  →  Det  N
VP  →  VP  NP

Det  →  the
N  →  dog
V  →  chased

S
→
NP
Det
N

→
the
dog
Register
S
NP
Det
the
N
dog
VP

Temporary Memory
S → NP VP
NP → Det N
VP → VP NP

Det → the
N → dog
V → chased

S
   ____________
   |
   NP
   |
   Det N
   |
   the dog
Register:

Temporary Memory:

S → NP VP
NP → Det N
VP → VP NP

Det → the
N → dog
V → chased

S

NP

Det

N

dog

VP
S → NP VP
NP → Det N
*VP → VP NP
Det → the
N → dog
N → cat
V → chased

S
  /   \
  NP   VP
     /   \
    Det N
       /   \
      the dog
Register
|
S
|
NP
|
Det
|
the
|
N
|
dog
|
VP
|
V
|

Temporary Memory

S → NP VP
NP → Det N
VP → VP NP

Det → the
N → dog
V → chased
<table>
<thead>
<tr>
<th>Register</th>
<th>Temporary Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>VP</td>
</tr>
<tr>
<td>NP</td>
<td>VP</td>
</tr>
<tr>
<td>Det</td>
<td>N VP</td>
</tr>
<tr>
<td>the</td>
<td>N VP</td>
</tr>
<tr>
<td>N</td>
<td>VP</td>
</tr>
<tr>
<td>dog</td>
<td>VP</td>
</tr>
<tr>
<td>chased</td>
<td>NP</td>
</tr>
</tbody>
</table>

- \(S \rightarrow \text{NP VP}\)
- \(\text{NP} \rightarrow \text{Det N}\)
- \(\text{VP} \rightarrow \text{VP NP}\)

- \(\text{Det} \rightarrow \text{the}\)
- \(\text{N} \rightarrow \text{dog}\)
- \(\text{N} \rightarrow \text{cat}\)
- \(*V \rightarrow \text{chased}\)

```
  S
   /\    /
  NP  VP
     /\    /
    Det N  V
       /\  /
      the dog
```
Register
S
NP
Det
the
N
dog
VP
V
chased

Temporary Memory
S \rightarrow NP \ VP
NP \rightarrow Det \ N
VP \rightarrow VP \ NP

Det \rightarrow the
N \rightarrow dog
N \rightarrow cat
V \rightarrow chased
Register

S
NP
Det
the
N
dog
VP
V
chased
NP

Temporary Memory

S → NP VP
NP → Det N
VP → VP NP

Det → the
N → dog
N → cat
V → chased
Register

S
NP
Det
the
N
dog
VP
chased
NP

Temporary Memory

S → NP VP
NP → Det N
VP → VP NP

Det → the
N → dog
V → chased

```
S
  /\  
 NP VP
 /\   /
Det N VP
 /\   /
the N VP
 /\   /
N VP
 /\   /
dog VP
 /\   /
chased NP
   /\   /
  the dog chased
```
The diagram represents the syntactic structure of the sentence "the dog chased". The rules for grammar are:

- $S \rightarrow NP \ VP$
- $*NP \rightarrow \text{Det} \ N$
- $VP \rightarrow VP \ NP$
- $\text{Det} \rightarrow \text{the}$
- $N \rightarrow \text{dog}$
- $N \rightarrow \text{cat}$
- $V \rightarrow \text{chased}$

The parse tree shows the structure of the sentence with the root $S$, a non-terminal node $NP$, and a non-terminal node $VP$. The subtree $NP$ contains a determiner $\text{Det}$, a noun $N$ with the value "the", and the noun $N$ with the value "dog". The subtree $VP$ contains a verb $V$ with the value "chased".
| Register | Temporary Memory |  |  |
|----------|------------------|  |  |
| S        |                  |  |  |
| NP       | VP               |  |  |
| Det      | N VP             |  |  |
| the      | N VP             |  |  |
| N        | VP               |  |  |
| dog      | VP               |  |  |
| VP       |                  |  |  |
| V        | NP               |  |  |
| chased   | NP               |  |  |
| NP       |                  |  |  |
| Det      | N                |  |  |
| the      | dog              |  |  |
| chased   | Det              |  |  |

S → NP VP
NP → Det N
VP → VP NP

Det → the
N → dog
N → cat
V → chased
Register  | Temporary Memory
---|---
S | S → NP VP
NP | NP → Det N
Det | VP → VP NP
the | *Det → the
N | N → dog
dog | N → cat
VP | V → chased
chased | V → chased
Register

Temporary Memory

S → NP VP
NP → Det N
VP → VP NP

Det → the
N → dog
N → cat
V → chased

S

NP

VP

N

V

NP

N

the

VP

N

the
Register  

S  
NP  
Det  
the  
N  
dog  
VP  
chased  
NP  
Det  
the  
N  
Temporary Memory  

S → NP VP  
NP → Det N  
VP → VP NP  

Det → the  
N → dog  
N → cat  
V → chased
Register
S
NP
Det
the
N
dog
VP
V
chased
NP
Det
the
N

Temporary Memory
S → NP VP
NP → Det N
VP → VP NP

Det → the
N → dog
N → cat
V → chased

S

NP

VP

Det
the
N
dog

VP

V

chased

NP

Det
the
N

N

the
dog
chased

Det
the
<table>
<thead>
<tr>
<th>Register</th>
<th>Temporary Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td></td>
</tr>
<tr>
<td>NP</td>
<td>VP</td>
</tr>
<tr>
<td>Det</td>
<td>N VP</td>
</tr>
<tr>
<td>the</td>
<td>N VP</td>
</tr>
<tr>
<td>N</td>
<td>VP</td>
</tr>
<tr>
<td>dog</td>
<td>VP</td>
</tr>
<tr>
<td>VP</td>
<td>VP</td>
</tr>
<tr>
<td>V</td>
<td>NP</td>
</tr>
<tr>
<td>chased</td>
<td>NP</td>
</tr>
<tr>
<td>NP</td>
<td></td>
</tr>
<tr>
<td>Det</td>
<td>N</td>
</tr>
<tr>
<td>the</td>
<td>N</td>
</tr>
<tr>
<td>N</td>
<td></td>
</tr>
<tr>
<td>cat</td>
<td></td>
</tr>
</tbody>
</table>

S → NP VP
NP → Det N
VP → VP NP
Det → the
N → dog
*N → cat
V → chased

```
S
  / \  /
NP   VP
  /   /
Det N V NP
  /   /
the dog chased Det N
    /   /
  the  the
```
Register
S
NP
Det
the
N
dog
VP
V
chased
NP
Det
the
N
cat

Temporary Memory
S → NP VP
NP → Det N
VP → VP NP
Det → the
N → dog
N → cat
V → chased

S
  └── NP  ┘── VP
       └── Det N  ┘── V
                     └── NP
                                 └── Det N
                                          └── the cat
One type of structure that the model thus far cannot produce is a *discontinuous constituent*:
To make the model more expressive, allow special rules affecting the temporary register:

\[
S \rightarrow NP \ VP \\
NP \rightarrow Pron \ldots Scomp
\]
To make the model more expressive, allow special rules affecting the temporary register:

\[
S \rightarrow NP \ VP \\
NP \rightarrow Pron. . . \ Scomp
\]

When a \ldots rule is used, the subsequent symbols go \textit{to the right of} the first symbol in memory:
To make the model more expressive, allow special rules affecting the temporary register:

\[
S \rightarrow NP \ VP \\
NP \rightarrow Pron. . . Scomp
\]

When a . . . rule is used, the subsequent symbols go *to the right of* the first symbol in memory:

\[
\text{Register} \mid \text{Memory}
\]
To make the model more expressive, allow special rules affecting the temporary register:

\[
S \rightarrow NP \ VP \\
NP \rightarrow Pron. . . Scomp
\]

When a . . . rule is used, the subsequent symbols go to the right of the first symbol in memory:

\[
\begin{array}{c|c}
\text{Register} & \text{Memory} \\
\hline
S & \end{array}
\]
To make the model more expressive, allow special rules affecting the temporary register:

\[
S \rightarrow NP \ VP \\
NP \rightarrow Pron. \ldots Scomp
\]

When a \ldots rule is used, the subsequent symbols go to the right of the first symbol in memory:

\[
\begin{array}{c|c}
\text{Register} & \text{Memory} \\
S & S
\end{array}
\]
To make the model more expressive, allow special rules affecting the temporary register:

\[
S \rightarrow \text{NP VP} \\
\text{NP} \rightarrow \text{Pron... Scomp}
\]

When a ... rule is used, the subsequent symbols go to the right of the first symbol in memory:

<table>
<thead>
<tr>
<th>Register</th>
<th>Memory</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NP</td>
<td>VP</td>
<td></td>
</tr>
</tbody>
</table>
To make the model more expressive, allow special rules affecting the temporary register:

\[
S \rightarrow NP \ VP \\
NP \rightarrow Pron...Scomp
\]

When a ... rule is used, the subsequent symbols go to the right of the first symbol in memory:

```
<table>
<thead>
<tr>
<th>Register</th>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>VP</td>
</tr>
<tr>
<td>NP</td>
<td>VP</td>
</tr>
</tbody>
</table>
```

\[
\text{S} \quad \text{NP} \quad \text{VP}
\]
To make the model more expressive, allow special rules affecting the temporary register:

\[
S \rightarrow \text{NP } \text{VP} \\
\text{NP} \rightarrow \text{Pron} \ldots \text{Scomp}
\]

When a \ldots rule is used, the subsequent symbols go to the right of the first symbol in memory:

<table>
<thead>
<tr>
<th>Register</th>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td></td>
</tr>
<tr>
<td>NP</td>
<td>VP</td>
</tr>
<tr>
<td>Pron</td>
<td>VP Scomp</td>
</tr>
</tbody>
</table>
To make the model more expressive, allow special rules affecting the temporary register:

\[ S \rightarrow NP \ VP \]
\[ NP \rightarrow Pron \ldots Scomp \]

When a \ldots rule is used, the subsequent symbols go to the right of the first symbol in memory:
In modern parlance, Yngve’s model is an incremental, top-down tree generator.
In modern parlance, Yngve’s model is an incremental, top-down tree generator.

While deep parts of the tree are being generated, upcoming constituents high in the tree have to be stored in memory.
In modern parlance, Yngve’s model is an *incremental, top-down* tree generator.

While deep parts of the tree are being generated, upcoming constituents high in the tree have to be stored in memory.

The number of symbols stored in temporary memory at any point in production of a sentence can be said to be the *depth* at that point.
The hypothesis

- Yngve wanted to identify the *temporary memory* of his model with the human cognitive resource of *working memory*.
The hypothesis

- Yngve wanted to identify the *temporary memory* of his model with the human cognitive resource of *working memory*.
- Since working memory is limited, sentences involving many items in the temporary memory should be hard to process.
Yngve implicitly proposed two predictions of his theory:

1. The greater the maximum depth of a sentence, the harder it should be to process (produce and/or comprehend).
Yngve implicitly proposed two predictions of his theory:

1. The greater the maximum depth of a sentence, the harder it should be to process (produce and/or comprehend).
2. When there are multiple alternative forms that can express the same meaning, the alternative with the smallest maximum depth should tend to be the most felicitous.
Yngve implicitly proposed two predictions of his theory:

1. The greater the maximum depth of a sentence, the harder it should be to process (produce and/or comprehend).
2. When there are multiple alternative forms that can express the same meaning, the alternative with the smallest maximum depth should tend to be the most felicitous.
3. Natural language grammars should tend to be structured so as to minimize depth.
Using multiple relative clauses in a sentence can make processing difficult:

This is the malt that the rat that the cat that the dog worried killed ate.
Using multiple relative clauses in a sentence can make processing difficult:

*This is the malt that the rat that the cat that the dog worried killed ate.*

It’s not the meaning of the sentence, or the use of relative clauses, that makes it hard:
Using multiple relative clauses in a sentence can make processing difficult:

This is the malt that the rat that the cat that the dog worried killed ate.

It’s not the meaning of the sentence, or the use of relative clauses, that makes it hard:

This is the malt that was eaten by the rat that was killed by the cat that was worried by the dog.
Using multiple relative clauses in a sentence can make processing difficult:

This is the malt that the rat that the cat that the dog worried killed ate.

It’s not the meaning of the sentence, or the use of relative clauses, that makes it hard:

This is the malt that was eaten by the rat that was killed by the cat that was worried by the dog.

What is hard about these sentences?
Center embedding and working memory

Register

the

Memory

the

N VP VP VP VP
this is the malt that was eaten by the rat that was killed by the cat that was worried by the dog.
Dem this is the malt that was eaten by the rat that was killed by the cat that was worried by the dog.

Right-embedding doesn’t have substantial temporary memory requirements.
In center-embedded structures, the model must “remember” which constituents have been started but are not yet finished.
In center-embedded structures, the model must “remember” which constituents have been started but are not yet finished.

In right-embedded structures, nothing of the sort needs to be remembered.
In center-embedded structures, the model must “remember” which constituents have been started but are not yet finished.

In right-embedded structures, nothing of the sort needs to be remembered.

That is, constructions with high maximum depth are hard to process.
The contents of temporary memory at any point can also be thought of as a set of “constituent expectations”:

\[
S \\
NP \quad VP \\
Dem \quad V \quad NP \\
this \quad is \quad Det \quad N \\
the \quad malt \quad RelPro \\
that \quad S \\
NP \quad VP \\
Det \quad N \\
the
\]
The contents of temporary memory at any point can also be thought of as a set of “constituent expectations”:  

```
S
   /   |
NP   VP
     /   |
Dem  this
    /   |
    V    is
     /   |
     Det  N
       /   |
       the  malt
         /   |
         RC   RelPro
          /   |
          that
            /   |
            NP
              /   |
              the

Register: the Memory: N VP
```
Preference to postpone “heavy” constituents until the end of the clause:
Preference to postpone “heavy” constituents until the end of the clause:

*Susan gave candy to the boy that visited his parents in New York last Christmas.*
Preference to postpone “heavy” constituents until the end of the clause:

Susan gave candy to the boy that visited his parents in New York last Christmas.

? Susan gave the boy that visited his parents in New York last Christmas candy.
Preference to postpone “heavy” constituents until the end of the clause:

Susan gave candy to the boy that visited his parents in New York last Christmas.

? Susan gave the boy that visited his parents in New York last Christmas candy.

The medial constituent has an additional memory penalty due to needing to remember the final constituent.
Preference to postpone “heavy” constituents until the end of the clause:

Susan gave candy to the boy that visited his parents in New York last Christmas.

Susan gave the boy that visited his parents in New York last Christmas candy.

The medial constituent has an additional memory penalty due to needing to remember the final constituent.

Hence, the “heavy-medial” variant has the greatest maximum memory requirement.
“Heavy Shift”

- Preference to postpone “heavy” constituents until the end of the clause:
  
  Susan gave candy to the boy that visited his parents in New York last Christmas.
  
  ? Susan gave the boy that visited his parents in New York last Christmas candy.

- The medial constituent has an additional memory penalty due to needing to remember the final constituent.

- Hence, the “heavy-medial” variant has the greatest maximum memory requirement.

- That is, when multiple forms are available for the same meaning, the “shallowest” alternative is preferred.
Consider the grammar of postmodified superlative adjectives such as *best... in the world*:
Consider the grammar of postmodified superlative adjectives such as *best... in the world:*

\[
\text{*the best in the world friend}
\]
Consider the grammar of postmodified superlative adjectives such as *best... in the world*:

*the best in the world friend
the best friend in the world
Consider the grammar of postmodified superlative adjectives such as *best... in the world*:

*the best in the world friend
the best friend in the world*

The discontinuous-constituent form used in English is less depth-intensive than the continuous-constituent form.
Consider the grammar of postmodified superlative adjectives such as *best... in the world*:

- *the best in the world friend*
- *the best friend in the world*

The discontinuous-constituent form used in English is less depth-intensive than the continuous-constituent form.

This type of evidence could be taken as support for Yngve’s hypothesis that grammars tend to avoid depth-intensive constructions.
Implications of a memory-based theory

- Yngve’s is a memory-based theory where the number of items in memory is crucial
Implications of a memory-based theory

- Yngve’s is a memory-based theory where the number of items in memory is crucial.
- In any such theory, *what counts as an item* is crucial.
- This is the problem of the units of representation.
Branching structure matters in memory-based theories.
Branching structure matters in memory-based theories. Ternary constructions introduce a memory load of 2:

```
X
O  O  O
[2] [1] [0]
```
Implications of a memory-based theory (2)

Branching structure matters in memory-based theories. Ternary constructions introduce a memory load of 2:

```
\[ \begin{array}{c}
  X \\
  O \quad O \\
  [2] \quad [1] \quad [0]
  \end{array} \]
```

Weakly equivalent binary constructions have load of only 1:

```
\[ \begin{array}{c}
  X \\
  O \quad Y \\
  [1] \\
  O \quad O \\
  [1] \quad [0]
  \end{array} \]
```
Weaknesses for Yngve’s theory

- As an incremental sentence processor, the model “over-commits” in many types of constructions.
- Left-branching constructions are pervasive in some languages (e.g., Japanese) and not necessarily hard.
Sometimes the model must commit to a structure before any definitive evidence for it has appeared:

```
S
   NP   VP
   |     |
  Pron  V
     |     |
   I    eat
```
Sometimes the model must commit to a structure before any definitive evidence for it has appeared:

```
S   S
  NP  VP   NP  VP
     |    |     |    |
     Pron  V   Pron  V  NP
         |    |         |    |
         I  I   I  I  I
         I  eat   I  eat
```
Sometimes the model must commit to a structure before any definitive evidence for it has appeared:
Overcommitment

Sometimes the model must commit to a structure before any definitive evidence for it has appeared:
Japanese and many other languages have pervasive left-embedding.
Japanese and many other languages have pervasive left-embedding.
Theories focusing on constituent expectations and incremental parsing choices/difficulty (Kimball, 1973; Gibson, 1991):

- What happened *yesterday*?

> *Martha expected that it would rain yesterday.*
Influence of Yngve on later work

Theories focusing on constituent expectations and incremental parsing choices/difficulty (Kimball, 1973; Gibson, 1991):

- What happened *yesterday*?

  *Martha expected that it would rain yesterday.*

- **RIGHT ASSOCIATION** (Kimball, 1973): terminal symbols (=words) optimally associate to lowest non-terminal node.
Theories of center-embedding difficulty have become more nuanced, as our understanding of data have improved (Gibson, 1991; Lewis, 1996; Gibson, 1998; Lewis and Vasishth, 2005):
Theories of center-embedding difficulty have become more nuanced, as our understanding of data have improved (Gibson, 1991; Lewis, 1996; Gibson, 1998; Lewis and Vasishth, 2005):

The fact that the employee who the manager hired __ stole office supplies worried the executive. (SC/RC)
Influence of Yngve on later work

- Theories of center-embedding difficulty have become more nuanced, as our understanding of data have improved (Gibson, 1991; Lewis, 1996; Gibson, 1998; Lewis and Vasishth, 2005):
  - The fact that the employee who the manager hired ___ stole office supplies worried the executive. (SC/RC)
  - # The executive who the fact that the employee stole office supplies worried ___ hired the manager. (RC/SC)
Gibson (1998, 2000): syntactic predictions are costly, but not all predictions are created equal, and keeping predictions over longer linear distances is costlier.
Gibson (1998, 2000): syntactic predictions are costly, but not all predictions are created equal, and keeping predictions over longer linear distances is costlier

- Sentential complements (SCs) predict just a *verb*
Gibson (1998, 2000): syntactic predictions are costly, but not all predictions are created equal, and keeping predictions over longer linear distances is costlier

- Sentential complements (SCs) predict just a verb
- Relative Clauses (RCs) predict both verb & wh-trace
Influence of Yngve on later work

- Gibson (1998, 2000): syntactic predictions are costly, but not all predictions are created equal, and keeping predictions over longer linear distances is costlier
  - Sentential complements (SCs) predict just a verb
  - Relative Clauses (RCs) predict both verb & wh-trace

The fact that the employee who the manager...
Influence of Yngve on later work

Gibson (1998, 2000): syntactic predictions are costly, but not all predictions are created equal, and keeping predictions over longer linear distances is costlier

- Sentential complements (SCs) predict just a verb
- Relative Clauses (RCs) predict both verb & wh-trace

The fact that the employee who the manager...
Gibson (1998, 2000): syntactic predictions are costly, but not all predictions are created equal, and keeping predictions over longer linear distances is costlier

- Sentential complements (SCs) predict just a *verb*
- Relative Clauses (RCs) predict both *verb* & *wh-trace*

The fact that the employee who the manager...
Influence of Yngve on later work

Gibson (1998, 2000): syntactic predictions are costly, but *not all predictions are created equal*, and *keeping predictions over longer linear distances is costlier*

- Sentential complements (SCs) predict just a *verb*
- Relative Clauses (RCs) predict both *verb* & *wh-trace*

The fact that the employee who the manager...
Influence of Yngve on later work

- Gibson (1998, 2000): syntactic predictions are costly, but *not all predictions are created equal*, and *keeping predictions over longer linear distances is costlier*
  - Sentential complements (SCs) predict just a *verb*
  - Relative Clauses (RCs) predict both *verb & wh-trace*

The fact that the employee who the manager...  

The executive who the fact that the employee...
Gibson (1998, 2000): syntactic predictions are costly, but not all predictions are created equal, and keeping predictions over longer linear distances is costlier

- Sentential complements (SCs) predict just a verb
- Relative Clauses (RCs) predict both verb & wh-trace

The fact that the employee who the manager...
Influence of Yngve on later work

- Gibson (1998, 2000): syntactic predictions are costly, but not all predictions are created equal, and keeping predictions over longer linear distances is costlier
  - Sentential complements (SCs) predict just a verb
  - Relative Clauses (RCs) predict both verb & wh-trace

The fact that the employee who the manager...

The executive who the fact that the employee...
Gibson (1998, 2000): syntactic predictions are costly, but not all predictions are created equal, and keeping predictions over longer linear distances is costlier

- Sentential complements (SCs) predict just a *verb*
- Relative Clauses (RCs) predict both *verb* & *wh-trace*

The fact that the employee who the manager...

The executive who the fact that the employee...
The mechanism for discontinuous constituents bears a close relationship to grammars with wrapping mechanisms (Pollard, 1984).

*It is true that I lied.*

\[
S_{\alpha\gamma\beta} \rightarrow NP_{\alpha|\beta} \ VP_{\gamma} \\
NP_{\alpha|\beta} \rightarrow NP_{\alpha} \ Scomp_{\beta}
\]
The empirical phenomenon of “heavy shift” has received considerable recent attention (Hawkins, 1994; Wasow, 2002)

Pauline gave to every child a beautiful
Influence of Yngve on later work

- The empirical phenomenon of “heavy shift” has received considerable recent attention (Hawkins, 1994; Wasow, 2002)

  *Pauline gave to every child a beautiful limited-edition book written by J.K. Rowling.*

- Semantic/pragmatic/discourse effects? (Arnold et al., 2000)
The empirical phenomenon of “heavy shift” has received considerable recent attention (Hawkins, 1994; Wasow, 2002)


Semantic/pragmatic/discourse effects? (Arnold et al., 2000)

How to measure constituent weight?
The empirical phenomenon of "heavy shift" has received considerable recent attention (Hawkins, 1994; Wasow, 2002)


Semantic/pragmatic/discourse effects? (Arnold et al., 2000)

How to measure constituent weight?

Are there processing costs associated (Arnold et al., 2004)
For Tuesday

Read Jurafsky & Narayanan, 1998


References II


