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

Roger Levy

UC San Diego
Department of Linguistics

LSA Summer Institute
6 July 2007
Computational Psycholinguistics

Psycholinguistics deals with the problem of how humans
- acquire
- represent
- comprehend
- produce
language.

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

- Overview the literature and major areas in which computational psycholinguistic research is carried out
- Gain experience in reading papers in this field
- Gain experience in working out the details of a model from the papers
- 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:

- **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
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
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
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
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
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
Introduction

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 an index card with the following info:

Name (optional)
School & Program/Department
Year/stage in program
Computational Linguistics background
Psycholinguistics background
Probability/Statistics background
Other courses you’re taking at LSA Summer Institute
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”
The model in more detail

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 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,
   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.
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,
   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.
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,
   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.
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,
   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.
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,
   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.
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,
   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.
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,
   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.
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,
   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.
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,
   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.
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

*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
- Det
- the
- N
- cat

**Temporary Memory**

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

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

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

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

Temporary Memory

*S → NP VP
*NP → Det N
*VP → VP NP

Det → the
N → dog
V → chased

S

NP

VP

Det N V NP

the dog chased Det N

the cat
Yngve 1960

The Model

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
The Model

Register
S
NP
Det
the
N
dog
VP
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
### 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*

### Diagram

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

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

Temporary Memory

\*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}
\*\text{V} \rightarrow \text{chased}
The Model

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
Computational Psycholinguistics: Day 1

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

Temporary Memory

*S → NP VP
*NP → Det N
*VP → VP NP

Det → the
N → dog
V → chased
Yngve 1960

The Model

Register

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

Temporary Memory

S → NP VP
NP → Det N
VP → VP NP

Det → the
N → dog
N → cat
V → chased
The Model

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 → Det N

VP → VP NP

Det → the

N → dog

N → cat

V → chased

NP → Det N

NP → Det N

VP → VP NP

Det → the

N → dog

N → cat

V → chased

NP → Det N

NP → Det N

NP → Det N

VP → VP NP

Det → the

N → dog

N → cat

V → chased

NP → Det N

NP → Det N

VP → VP NP

Det → the

N → dog

N → cat

V → chased

NP → Det N

NP → Det N

VP → VP NP

Det → the

N → dog

N → cat

V → chased

NP → Det N

NP → Det N

VP → VP NP

Det → the

N → dog

N → cat

V → chased

NP → Det N

NP → Det N

VP → VP NP

Det → the

N → dog

N → cat

V → chased

NP → Det N

NP → Det N

VP → VP NP

Det → the

N → dog

N → cat

V → chased

NP → Det N

NP → Det N

VP → VP NP

Det → the

N → dog

N → cat

V → chased

NP → Det N

NP → Det N

VP → VP NP

Det → the

N → dog

N → cat

V → chased

NP → Det N

NP → Det N

VP → VP NP

Det → the

N → dog

N → cat

V → chased

NP → Det N

NP → Det N

VP → VP NP

Det → the

N → dog

N → cat

V → chased

NP → Det N

NP → Det N

VP → VP NP

Det → the

N → dog

N → cat

V → chased

NP → Det N

NP → Det N

VP → VP NP

Det → the

N → dog

N → cat

V → chased

NP → Det N

NP → Det N

VP → VP NP

Det → the

N → dog

N → cat

V → chased

NP → Det N

NP → Det N

VP → VP NP

Det → the

N → dog

N → cat

V → chased

NP → Det N

NP → Det N

VP → VP NP

Det → the

N → dog

N → cat

V → chased

NP → Det N

NP → Det N

VP → VP NP

Det → the

N → dog

N → cat

V → chased

NP → Det N

NP → Det N

VP → VP NP

Det → the

N → dog

N → cat

V → chased

NP → Det N

NP → Det N

VP → VP NP

Det → the

N → dog

N → cat

V → chased

NP → Det N

NP → Det N

VP → VP NP

Det → the

N → dog

N → cat

V → chased

NP → Det N

NP → Det N

VP → VP NP

Det → the

N → dog

N → cat

V → chased

NP → Det N

NP → Det N

VP → VP NP

Det → the

N → dog

N → cat

V → chased

NP → Det N

NP → Det N

VP → VP NP

Det → the

N → dog

N → cat

V → chased

NP → Det N

NP → Det N

VP → VP NP

Det → the

N → dog

N → cat

V → chased
Computational Psycholinguistics: Day 1

Yngve 1960

The Model

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

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

*S → NP VP
*NP → Det N
*VP → VP NP

*Det → the
*N → dog
*N → cat
*V → chased
The Model

Register:
- S
- NP
- Det
- the N
- VP
- dog
- V
- chased

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

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

Diagram:
```
S
  / \  
|   |   |
NPVP
  /  \  /
|   |   |
DetNVP
  /  \  /
|   |   |
theNVP
  /  \  /
|   |   |
thedogVP
  /  \  /
|   |   |
chasedNP
  /  \  /
|   |   |
thedetN
  /  \  /
|   |   |
thethecat
```
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

Diagram:

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

Temporary Memory

\[ S \rightarrow NP \ VP \]
\[ NP \rightarrow Det \ N \]
\[ VP \rightarrow VP \ NP \]
\[ Det \rightarrow the \]
\[ N \rightarrow dog \]
\[ N \rightarrow cat \]
\[ V \rightarrow chased \]

Diagram:

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

S
NP
Det
the
N
dog
VP
V
chased

Temporary Memory

*S → NP VP
*NP → Det N
*VP → VP NP

Det → the
N → dog
*V → chased
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

Diagram:

S
  /\  
NP    VP
 / \   / \  
Det N VP NP
 / \ / \ / \ 
the dog chased the cat
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

The Model

*S → NP VP
*NP → Det N
*VP → VP NP

*Det → the
*N → dog
*N → cat
*V → chased
The Model

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
Register

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

Temporary Memory

*S → NP VP
*N → Det N
*VP → VP NP

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

Det

N

the
dog

VP

chased

NP

Det

N

the
cat
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

\[ S \rightarrow NP \quad VP \rightarrow VP \quad NP \rightarrow Det \quad N \rightarrow \text{dog} \quad N \rightarrow \text{cat} \quad V \rightarrow \text{chased} \]
Register
S
NP
Det
the
N
dog
VP
V
chased
NP
Det
the
N
cat

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, N, VP, chased, NP
Temporary Memory:
- *S → NP VP
- *NP → Det N
- *VP → VP NP

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

Diagram:
```
  S
 /\  
/   \\
NP /     \\VP
 Det  N  V  NP
   \  /\  /
    the dog chased
     \  /\  /
N  the N  N
     \  /
      cat
```
The Model

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 VP
N VP
VP
V
chased
NP
Det
the
N
cat

the
dog
chased
the
cat
Discontinuous constituents

One type of structure that the model thus far cannot produce is a *discontinuous constituent*:

\[
\begin{array}{c}
S \\
\searrow \\
NP \\
\searrow \\
Pron \quad V \\
\searrow \\
\quad Adj \\
\searrow \\
\quad true \\
\searrow \\
Scomp \\
\searrow \\
\quad that \\
\searrow \\
\quad NP \\
\searrow \\
Pron \\
\searrow \\
\quad V \\
\searrow \\
\quad lied
\end{array}
\]
Discontinuous constituents (2)

To make the model more expressive, allow special rules affecting the temporary register:

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

When a ... rule is used, the subsequent symbols go to the right of the first symbol in memory:
Discontinuous constituents (2)

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{align*}
\text{Register} & | \quad \text{Memory} \\
S & | \quad NP \quad VP \\
NP & | \quad VP \quad Scomp \\
\text{Pron} & | \quad \text{VP} \quad \text{Scomp}
\end{align*}
Discontinuous constituents (2)

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{align*}
\text{Register} & | \quad \text{Memory} \\
S & | S \\
NP & | NP \ \ VP \\
Pron & | Pron \ \ VP \ \ Scomp
\end{align*}
Discontinuous constituents (2)

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:

\[
\begin{array}{c|c}
\text{Register} & \text{Memory} \\
S & \\
NP & VP \\
Pron & VP Scomp
\end{array}
\]
Discontinuous constituents (2)

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:
Discontinuous constituents (2)

To make the model more expressive, allow special rules affecting the temporary register:

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

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

\[
\begin{array}{c|c}
\text{Register} & \text{Memory} \\
S & \\
NP & VP \\
Pron & VP \text{ Scomp}
\end{array}
\]
Discontinuous constituents (2)

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:

\[
\text{Register} \quad \text{Memory} \\
S \quad S \\
NP \quad NP \\
Pron \quad Pron \\
VP \quad VP \ Scomp
\]
Discontinuous constituents (2)

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 \textit{to the right of} the first symbol in memory:

\begin{tabular}{c|c}
Register & Memory \\
S & \\
NP & VP \\
Pron & VP Scomp
\end{tabular}
Discontinuous constituents (2)

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 \textit{to the right of} the first symbol in memory:

\[
\begin{array}{c|c}
\text{Register} & \text{Memory} \\
S & S \\
NP & VP \\
Pron & VP Scomp
\end{array}
\]
Model summary

- 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.
Model summary

- 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.
Model summary

- 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*.
- Since working memory is limited, sentences involving many items in the temporary memory should be hard to process.
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.
The hypothesis (2)

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.
The hypothesis (2)

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.
The hypothesis (2)

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.
Center embedding

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

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

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?
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 | Memory
the | N VP VP VP VP
Yngve's hypothesis

S
NP VP
Dem V NP
this is Det N RC
the malt RelPro S
that VP
V Part PP
was eaten P NP
by Det N RC
the rat RelPro S
that VP
V Part PP
was killed P NP
by Det N RC
the cat RelPro S
that VP
V Part PP
was worried P NP
by Det N
the dog
Right-embedding doesn’t have substantial temporary memory requirements.
Only center embedding fills temporary memory

- 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.
Only center embedding fills temporary memory

- 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.
Only center embedding fills temporary memory

- 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.
A note: temporary memory as “constituent expectations”

The contents of temporary memory at any point can also be thought of as a set of “constituent expectations”:

```
S
  /\  \
 NP  VP
  |  /
 Dem V  NP
  | /
 this is Det N RC
  | |
  the malt RelPro
  |  \\  \\  \\  \
  that S VP
  |  |
  Det N
  |  |  \
  the
```
A note: temporary memory as “constituent expectations”

The contents of temporary memory at any point can also be thought of as a set of “constituent expectations”:

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

Register Memory
the N VP
  |   |
the
“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.
“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.

  $\text{? } Susan$ 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.
“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.
Yngve’s hypothesis

“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
“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.
“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
Discontinuous constituents

▶ 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.
Discontinuous constituents

- 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.
Discontinuous constituents

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.
Discontinuous constituents

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.
Discontinuous constituents

- 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
- In any such theory, *what counts as an item* is crucial
- This is the problem of the units of *representation*
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*.
Implications of a memory-based theory (2)

Branching structure matters in memory-based theories.

Ternary constructions introduce a memory load of 2:

```
X
O   O   O
[2] [1] [0]
```

Weakly equivalent binary constructions have load of only 1:

```
X
O   Y
[1] [1] [0]
```

Implications of a memory-based theory (2)

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

```
X
O O O
[2] [1] [0]
```

Weakly equivalent binary constructions have load of only 1:

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

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

```
X
O O O
[2] [1] [0]
```

Weakly equivalent binary constructions have load of only 1:

```
X
O Y
[1] [1]
O O
[1] [0]
```
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
Overcommitment

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:

```
S             S             S
|              |              |
NP  VP         NP  VP         NP  VP
|               |               |
Pron  V         Pron  V  NP    Pron  V  NP  PP
|       |          |       |              |       |
\    /  \      /  \          /  \      /  \  /  \\
/    \  /      /  \          /  \      /  \  /  \\
I   eat  I   eat          I   eat
```
Overcommitment

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:

```
S
  NP  VP
    Pron  V
      /    /  
     /    /   
    I    I    I
   /    /    
  /    /     
I    I      
/    /      
/    /       
I    I        
I    I          
I    eat           
I    I              
I    I                
I    I                  
I    /                    
S
  NP  VP
    Pron  V
      /    /  
     /    /   
    I    I    I
   /    /    
  /    /     
I    I      
I    eat
```
Left-embedding

Japanese and many other languages have pervasive left-embedding.

\[
\text{S} \quad \text{CC} \quad \text{V} \\
\text{S} \quad \text{Comp} \quad \text{omou} \\
\text{S} \quad \text{CC} \quad \text{NP} \quad \text{V} \\
\text{NP} \quad \text{Comp} \quad \text{PropN} \quad \text{ittta} \\
\text{propN} \quad \text{to} \quad \text{John-ga} \\
\text{propN} \quad \text{kita} \\
\text{Mary-ga} \quad \text{arrived} \\
\text{Mary} 
\]
Left-embedding

Japanese and many other languages have pervasive left-embedding.
Influence of Yngve on later work

- Theories focusing on constituent expectations and incremental parsing choices/difficulty (Kimball, 1973; Gibson, 1991)
- Theories of center-embedding difficulty have become more nuanced, as our understanding of data have improved (Gibson, 1991; Lewis, 1996):
  - The fact that the employee who the manager hired stole office supplies worried the executive.
  - The executive who the fact that the employee stole office supplies worried hired the manager.
- The mechanism for discontinuous constituents bears a close relationship to grammars with wrapping mechanisms (Pollard, 1984)
- The empirical phenomenon of “heavy shift” has received considerable recent attention (Hawkins, 1994; Wasow, 2002)
Influence of Yngve on later work

- Theories focusing on constituent expectations and incremental parsing choices/difficulty (Kimball, 1973; Gibson, 1991)
- Theories of center-embedding difficulty have become more nuanced, as our understanding of data have improved (Gibson, 1991; Lewis, 1996):
  - The fact that the employee who the manager hired stole office supplies worried the executive.
  - # The executive who the fact that the employee stole office supplies worried hired the manager.
- The mechanism for discontinuous constituents bears a close relationship to grammars with wrapping mechanisms (Pollard, 1984)
- The empirical phenomenon of “heavy shift” has received considerable recent attention (Hawkins, 1994; Wasow, 2002)
Influence of Yngve on later work

- Theories focusing on constituent expectations and incremental parsing choices/difficulty (Kimball, 1973; Gibson, 1991)
- Theories of center-embedding difficulty have become more nuanced, as our understanding of data have improved (Gibson, 1991; Lewis, 1996):
  - The fact that the employee who the manager hired stole office supplies worried the executive.
  - # The executive who the fact that the employee stole office supplies worried hired the manager.
- The mechanism for discontinuous constituents bears a close relationship to grammars with wrapping mechanisms (Pollard, 1984)
- The empirical phenomenon of “heavy shift” has received considerable recent attention (Hawkins, 1994; Wasow, 2002)
Influence of Yngve on later work

- Theories focusing on constituent expectations and incremental parsing choices/difficulty (Kimball, 1973; Gibson, 1991)
- Theories of center-embedding difficulty have become more nuanced, as our understanding of data have improved (Gibson, 1991; Lewis, 1996):
  - The fact that the employee who the manager hired stole office supplies worried the executive.
  - # The executive who the fact that the employee stole office supplies worried hired the manager.
- The mechanism for discontinuous constituents bears a close relationship to grammars with wrapping mechanisms (Pollard, 1984)
- The empirical phenomenon of “heavy shift” has received considerable recent attention (Hawkins, 1994; Wasow, 2002)
Influence of Yngve on later work

- Theories focusing on constituent expectations and incremental parsing choices/difficulty (Kimball, 1973; Gibson, 1991)
- Theories of center-embedding difficulty have become more nuanced, as our understanding of data have improved (Gibson, 1991; Lewis, 1996):
  - The fact that the employee who the manager hired stole office supplies worried the executive.
  - # The executive who the fact that the employee stole office supplies worried hired the manager.
- The mechanism for discontinuous constituents bears a close relationship to grammars with wrapping mechanisms (Pollard, 1984)
- The empirical phenomenon of “heavy shift” has received considerable recent attention (Hawkins, 1994; Wasow, 2002)
Influence of Yngve on later work

- Theories focusing on constituent expectations and incremental parsing choices/difficulty (Kimball, 1973; Gibson, 1991)
- Theories of center-embedding difficulty have become more nuanced, as our understanding of data have improved (Gibson, 1991; Lewis, 1996):
  - The fact that the employee who the manager hired stole office supplies worried the executive.
  - # The executive who the fact that the employee stole office supplies worried hired the manager.
- The mechanism for discontinuous constituents bears a close relationship to grammars with wrapping mechanisms (Pollard, 1984)
- The empirical phenomenon of “heavy shift” has received considerable recent attention (Hawkins, 1994; Wasow, 2002)
For Tuesday

Read Jurafsky 1996, *focusing on Section 4.*
Yngve's hypothesis

References


