Chapter 14, Sections 14.1-14.4

Long Distance Dependencies
Overview

• Some examples of the phenomenon
• What is new and different about it
• Brief sketch of the TG approach
• Broad outlines of our approach
• More details of our approach next time
Examples

• *wh*-questions:
  
  *What did you find?*
  
  *Tell me who you talked to*

• relative clauses:
  
  *the item that I found*
  
  *the guy who(m) I talked to*

• topicalization:
  
  *The manual, I can’t find*
  
  *Chris, you should talk to.*

• *easy*-adjectives:
  
  *My house is easy to find.*
  
  *Pat is hard to talk to.*
What these have in common

• There is a ‘gap’: nothing following *find* and *to*, even though both normally require objects.

• Something that fills the role of the element missing from the gap occurs at the beginning of the clause.

• We use topicalization and *easy*-adjectives to illustrate:

  *The* manual, *I can’t find* _____

  *Chris* is easy to talk to to _____
Gaps and their fillers can be far apart:

• *The solution to this problem*, Pat said that someone claimed you thought I would never find_____.

• *Chris is easy to consider it impossible for anyone but a genius to try to talk to*****.*

♫ That’s why we call them “long distance dependencies”
Fillers often have syntactic properties associated with their gaps

Him, I haven’t met___.

*He, I haven’t met___.

The scissors, Pat told us ____ were missing.

*The scissors, Pat told us ____ was missing.

On Pat, you can rely___.

*To Pat, you can rely___.

© 2003 CSLI Publications
LDDs in TG

• These were long thought to constitute the strongest evidence for transformations.

• They were handled in TG by moving the filler from the gap position.

• Case, agreement, preposition selection could apply before movement.
A big debate about LDDs in TG

• Does long-distance movement take place in one fell swoop or in lots of little steps?

Swooping

Looping
Looping is now generally accepted in TG

- Various languages show morphological marking on the verbs or complementizers of clauses between the filler and the gap.
- Psycholinguistic evidence indicates increased processing load in the region between filler and gap.
- This opens the door to non-transformational analyses, in which the filler-gap dependency is mediated by local information passing.
Very Rough Sketch of Our Approach

• A feature GAP records information about a missing constituent.
• The GAP value is passed up the tree by a new principle.
• A new grammar rule expands S as a filler, followed by another S whose GAP value matches the filler.
• Caveat: Making the details of this general idea work involves several complications.
The Feature GAP

- Like valence features and ARG-ST, GAP’s value is a list of feature structures (often empty).
- Subject gaps are introduced by a lexical rule we’ll talk about next time.
- Non-subject gaps are introduced by revising the Argument Realization Principle.
The Revised ARP

\[
\begin{bmatrix}
\text{SYN} & \text{VAL} & \text{SPR} & A \\
\text{GAP} & \text{COMPS} & B \ominus C
\end{bmatrix}
\]

\[
\begin{bmatrix}
\text{ARG-ST} & A \oplus B
\end{bmatrix}
\]

- \(\ominus\) is a kind of list subtraction, but:
  - it’s not always defined, and
  - when defined, it’s not always unique

- The ARP now says the non-SPR arguments are distributed between COMPS and GAP.
A Word with a Non-Empty GAP Value

\[
\begin{align*}
\langle \text{hand} , & \quad \text{word} \rangle \\
\text{SYN} & \quad \left[ \begin{array}{l}
\text{HEAD} \\
\text{VAL} \\
\text{GAP}
\end{array} \right] \quad \left[ \begin{array}{l}
[\text{FORM } \text{fin}] \\
[\text{SPR} \langle 1 \rangle ] \\
[\text{COMPS} \langle 3 \text{PP[to]} \rangle ] \\
[\text{GAP} \langle 2 \text{NP[acc]} \rangle ]
\end{array} \right] \\
\text{ARG-ST} & \quad \left[ \begin{array}{l}
\text{CASE} \\
\text{AGR}
\end{array} \right] \quad \langle \begin{array}{l}
[\text{nom}] \\
[\text{non-3sing}] \\
[1], [2], [3]
\end{array} \rangle
\end{align*}
\]
How We Want GAP to Propagate
What We Want the GAP Propagation Mechanism to Do

• Pass any GAP values from daughters up to their mothers,
• except when the filler is found.
• For topicalization, we can write the exception into the grammar rule, but
• For *easy*-adjectives, the NP that corresponds to the gap is the subject, which is introduced by the Head-Specifier Rule.
• Since specifiers are not generally gap fillers, we can’t write the gap-filling into the HSR.
Our Solution to this Problem

• For *easy*-adjectives, we treat the adjective formally as the filler, marking its SPR value as coindexed with its GAP value.

• We use a feature STOP-GAP to trigger the emptying of the GAP list.
  • STOP-GAP stops gap propagation
  • *easy*-adjectives mark STOP-GAP lexically
  • a new grammar rule, the Head-Filler Rule mentions STOP-GAP
The GAP Principle

A local subtree $\Phi$ satisfies the GAP Principle with respect to a headed rule $\rho$ if and only if $\Phi$ satisfies:

$$[\text{GAP } (A_1 \oplus \ldots \oplus A_n) \ominus A_0]$$
How does STOP-GAP work?

- STOP-GAP is empty almost everywhere
- When a gap is filled, STOP-GAP is nonempty, and its value is the same as the gap being filled.
- This blocks propagation of that GAP value, so gaps are only filled once.
- The nonempty STOP-GAP values come from two sources:
  - a stipulation in the Head-Filler Rule
  - lexical entries for easy-adjectives
- No principle propagates STOP-GAP
The Head-Filler Rule

\[ \text{[phrase]} \rightarrow 1\text{[GAP }\langle\rangle\text{]} \]

- This only covers gap filling in finite Ss
- The filler has to be identical to the GAP value
- The STOP-GAP value is also identical
- The GAP Principle ensures that the mother’s GAP value is the empty list
Gap Filling with *easy*-Adjectives

Because STOP-GAP and GAP have the same value, that value will be subtracted from the mother’s GAP value.

The first argument is coindexed with the GAP value, accounting for the interpretation of the subject as the filler.
A Tree for *easy to talk to***

\[
\begin{align*}
&\text{A} \\
&\begin{cases}
\text{VAL} & \begin{cases}
\text{SPR} & \langle [2NP_i] \rangle \\
\text{COMPS} & \langle [3] \rangle
\end{cases} \\
\text{GAP} & \langle \rangle \\
\text{STOP-GAP} & \langle [1] \rangle
\end{cases}
\end{align*}
\]
Lexical entries with nonempty STOP-GAP values (like *easy*) are rare, so STOP-GAP is default empty in the lexicon.

HeadSpecifier and Head-Modifier rules need to say [STOP-GAP < >]

Lexical rules preserve STOP-GAP values.
GAP Housekeeping

• The initial symbol must say [GAP < >]. Why?
GAP Housekeeping

- The initial symbol must say [GAP < >]. Why?
- To block *Pat found* and *Chris talked to* as stand-alone sentences.
GAP Housekeeping

• The initial symbol must say [GAP < >]. Why?
  • To block *Pat found and *Chris talked to as stand-alone sentences.
• The Imperative Rule must propagate GAP values. Why?
GAP Housekeeping

• The initial symbol must say [GAP < >]. Why?
  • To block *Pat found and *Chris talked to as stand-alone sentences.

• The Imperative Rule must propagate GAP values. Why?
  • It’s not a headed rule, so the effect of the GAP Principle must be replicated
  • Imperatives can have gaps: This book, put on the top shelf!
Sentences with Multiple Gaps

• Famous examples:
  
  *This violin, sonatas are easy to play___ on____.
  *Sonatas, this violin is easy to play___ on____.

• Our analysis gets this:
  
  • The subject of easy is coindexed with the first element of the GAP list.
  • The Head-Filler rule only allows one GAP remaining.
  • There are languages that allow multiple gaps more generally.