Anaphora Resolution

Spring 2010, UCSC – Adrian Brasoveanu

[Slides based on various sources, collected over a couple of years and repeatedly modified – the work required to track them down & list them would take too much time at this point. Please email me (abrsvn@gmail.com) if you can identify particular sources.]

The Plan

Introduce and compare 3 algorithms for anaphora resolution:

- Hobbs 1978
- Lappin and Leass 1994
- Centering Theory

Hobbs 1978

- *Hobbs, Jerry R., 1978, ``Resolving Pronoun References'', Lingua, Vol. 44, pp. 311-338.*
  - *Also in Readings in Natural Language Processing, B. Grosz, K. Sparck-Jones, and B. Webber, editors, pp. 339-352, Morgan Kaufmann Publishers, Los Altos, California.*

Hobbs 1978

- Hobbs (1978) proposes an algorithm that searches parse trees (i.e., basic syntactic trees) for antecedents of a pronoun.
  - starting at the NP node immediately dominating the pronoun
  - in a specified search order
  - looking for the first match of the correct gender and number
- Idea: discourse and other preferences will be approximated by search order.

Hobbs's point

... *the naïve approach is quite good. Computationally speaking, it will be a long time before a semantically based algorithm is sophisticated enough to perform as well, and these results set a very high standard for any other approach to aim for.*

Yet there is every reason to pursue a semantically based approach. The naïve algorithm does not work. Any one can think of examples where it fails. In these cases it not only fails; it gives no indication that it has failed and offers no help in finding the real antecedent. (p. 345)

Hobbs 1978

- This simple algorithm has become a baseline: more complex algorithms should do better than this.
- Hobbs distance: i^{th} candidate NP considered by the algorithm is at a Hobbs distance of i.

A parse tree

The castle in Camelot remained the residence of the king until 536 when he moved it to London.

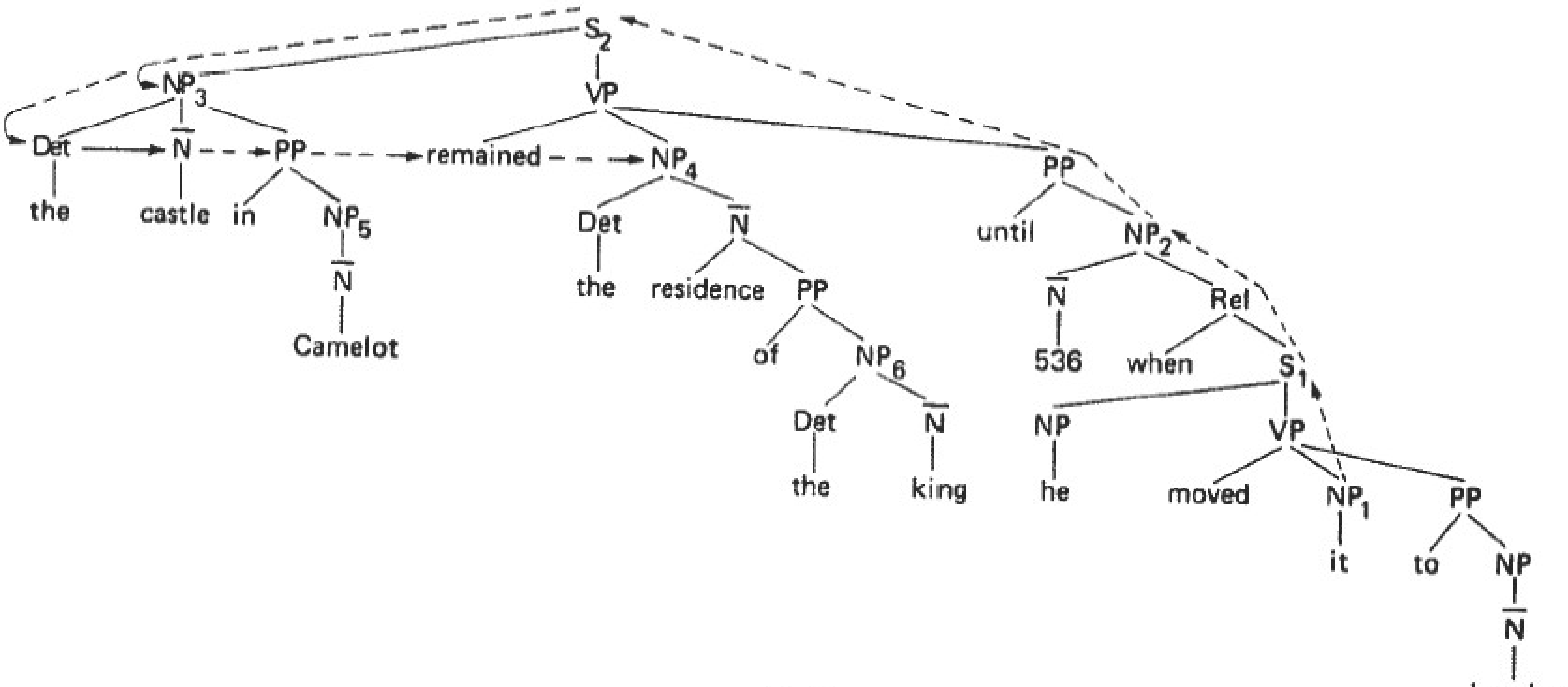


Fig. 2.

Multiple parse trees

Because it assumes parse trees, such an algorithm is inevitably dependent on one's theory of grammar.

1. Mr. Smith saw a driver in his truck.
2. Mr. Smith saw a driver of his truck.

"his" may refer to the driver in 1, but not 2.

- different parse trees explain the difference:
  - in 1, if the PP is attached to the VP, "his" can refer back to the driver;
  - in 2, the PP is obligatorily attached inside the NP, so "his" cannot refer back to the driver.

Hobbs's "Naïve" Algorithm

1. Begin at the NP immediately dominating the pronoun.
2. Go up tree to first NP or S encountered.
   - Call node X, and path to it, p.
   - Search left-to-right below X and to left of p, proposing any NP node which has an NP or S between it and X.
3. If X is highest S node in sentence,
   - Search previous trees, in order of recency, left-to-right, breadth-first, proposing NPs encountered.
4. Otherwise, from X, go up to first NP or S node encountered,
   - Call this X, and path to it p.
5. If X is an NP, and p does not pass through an N-bar that X immediately dominates, propose X.
6. Search below X, to left of p, left-to-right, breadth-first, proposing NP encountered.
7. If X is an S, search below X to right of p, left-to-right, breadth-first, but not going through any NP or S, proposing NP encountered.
8. Go to 2.

Add some hacks / heuristics

- Add “simple” selectional restrictions, e.g.:
  - dates can’t move
  - places can’t move
  - large fixed objects can’t move
- For “they”, in addition to accepting plural NPs, collects selectionally compatible entities (somehow), e.g., conjoined NPs.
- Assume some process that recovers elided constituents and inserts them in the tree.

Example:

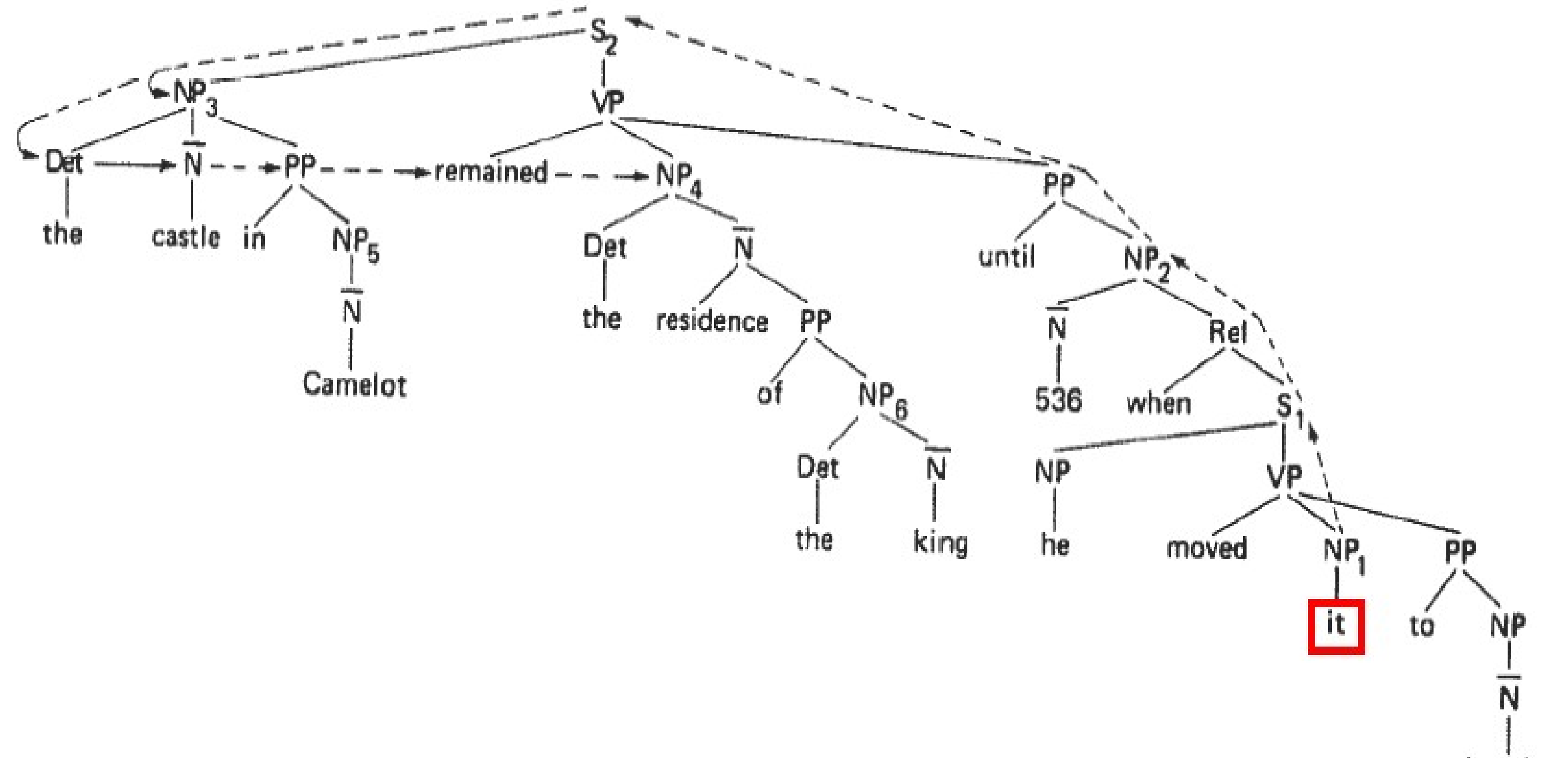


Fig. 2.

Let's try to find the referent for "it".

Hobbs’s “Naïve” Algorithm

1. Begin at the NP immediately dominating the pronoun.
2. Go up tree to first NP or S encountered.
   - Call node X, and path to it, p.
   - Search left-to-right below X and to left of p, proposing any NP node which has an NP or S between it and X.
3. If X is highest S node in sentence,
   - Search previous trees, in order of recency, left-to-right, breadth-first, proposing NPs encountered.
4. Otherwise, from X, go up to first NP or S node encountered,
   - Call this X, and path to it p.
5. If X is an NP, and p does not pass through an N-bar that X immediately dominates, propose X.
6. Search below X, to left of p, left-to-right, breadth-first, proposing NP encountered.
7. If X is an S, search below X to right of p, left-to-right, breadth-first, but not going through any NP or S, proposing NP encountered.
8. Go to 2.

Example:

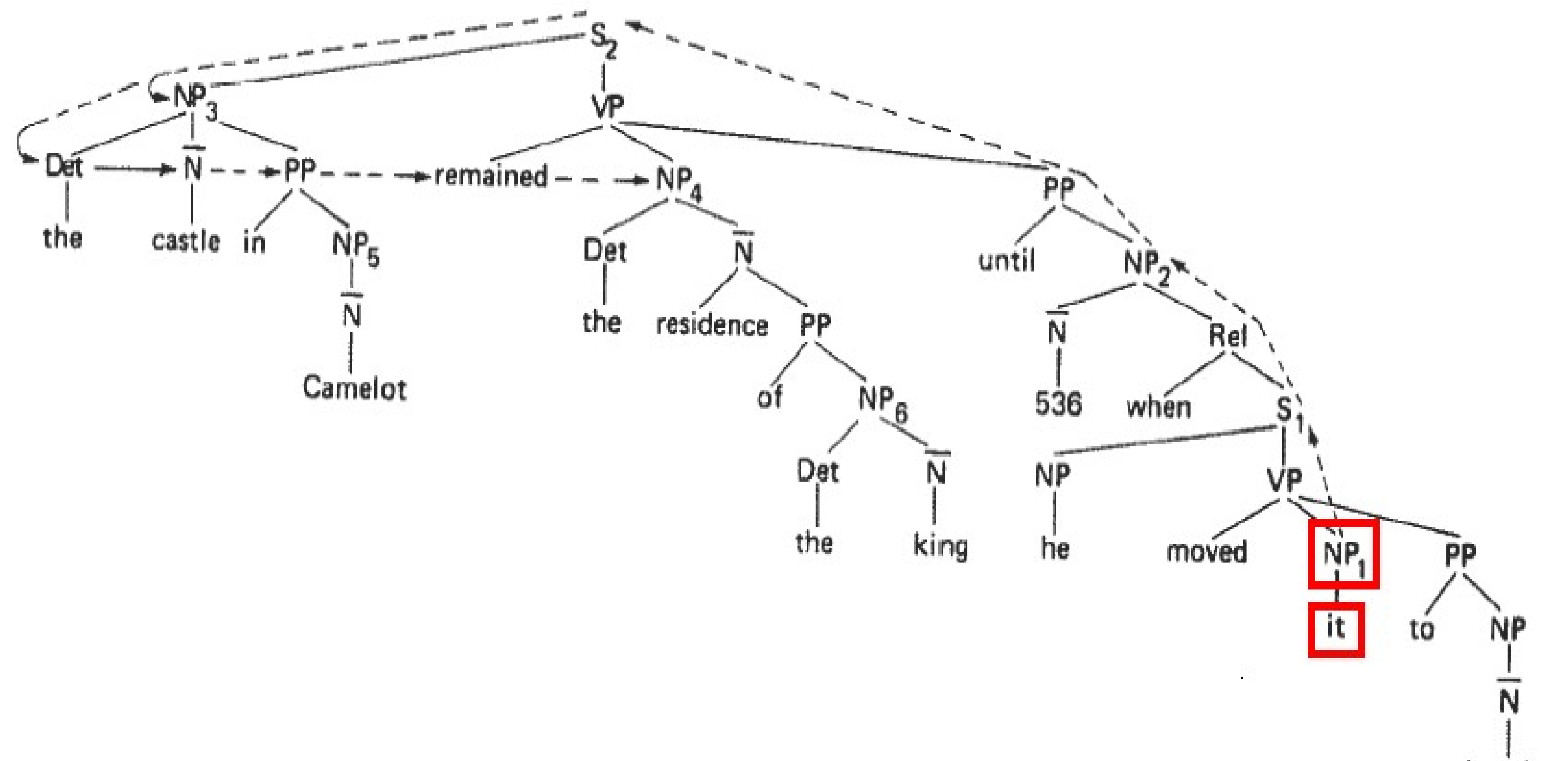


Fig. 2.

Begin at the NP immediately dominating the pronoun.

Hobbs's "Naïve" Algorithm

1. Begin at the NP immediately dominating the pronoun.
2. Go up tree to first NP or S encountered.
   - Call node X, and path to it, p.
   - Search left-to-right below X and to left of p, proposing any NP node which has an NP or S between it and X.
3. If X is highest S node in sentence,
   - Search previous trees, in order of recency, left-to-right, breadth-first, proposing NPs encountered.
4. Otherwise, from X, go up to first NP or S node encountered,
   - Call this X, and path to it p.
5. If X is an NP, and p does not pass through an N-bar that X immediately dominates, propose X.
6. Search below X, to left of p, left-to-right, breadth-first, proposing NP encountered.
7. If X is an S, search below X to right of p, left-to-right, breadth-first, but not going through any NP or S, proposing NP encountered.
8. Go to 2.

Example:

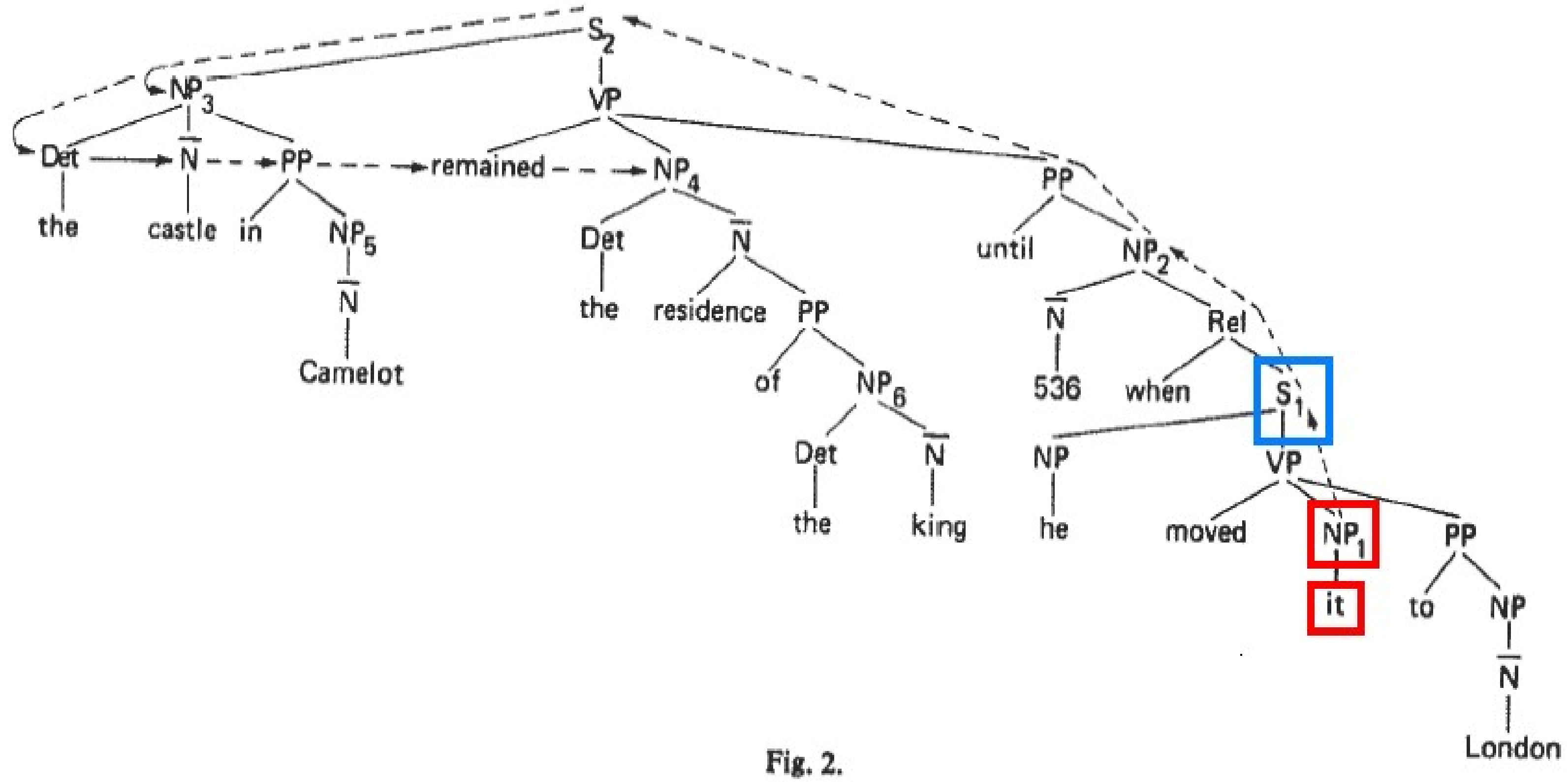


Fig. 2.

Go up tree to first NP or S encountered. Call node X, and path to it, p. Search left-to-right below X and to left of p, proposing any NP node which has an NP or S between it and X.

Example:

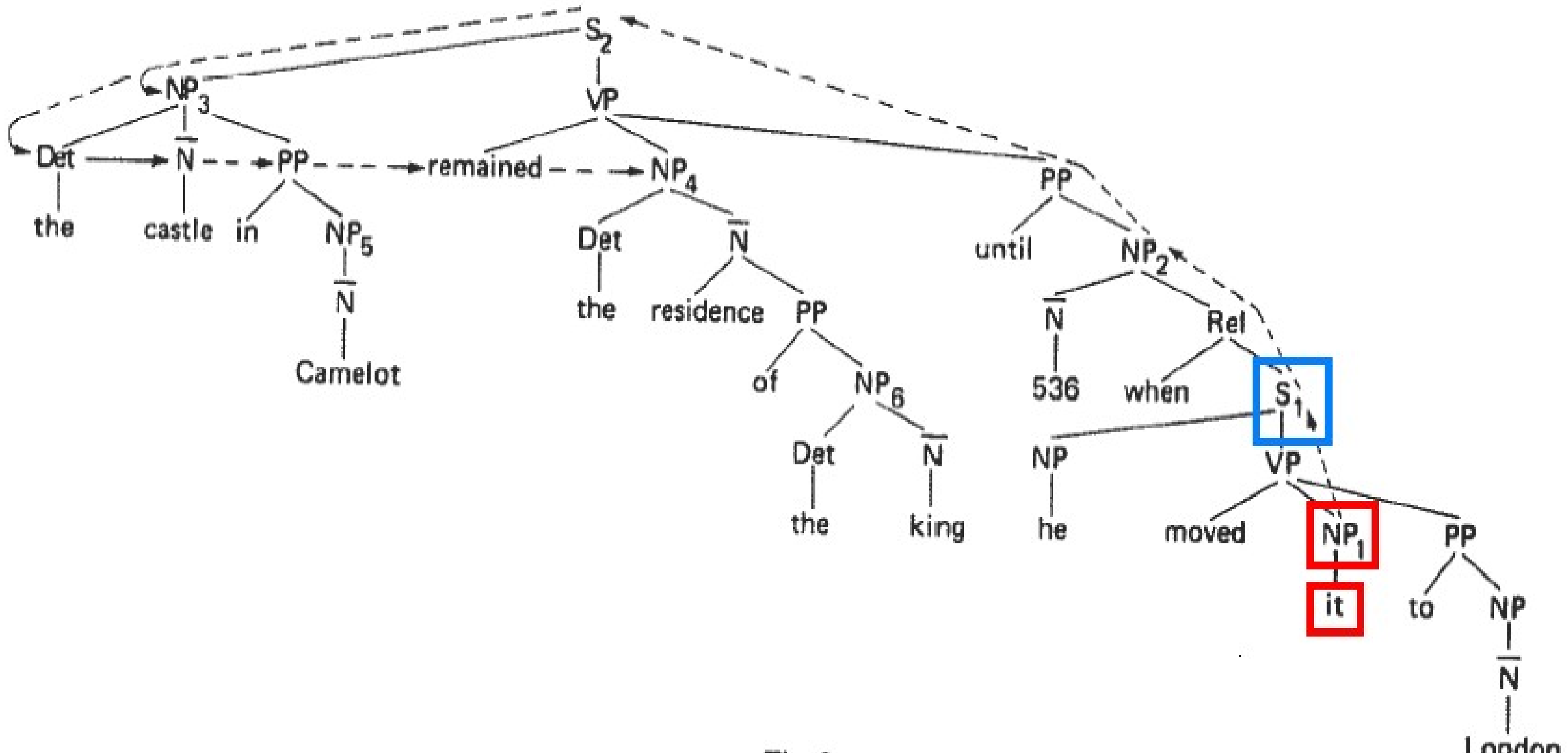


Fig. 2.

S1: search yields no candidate. Go to next step of the algorithm.

Hobbs's "Naïve" Algorithm

1. Begin at the NP immediately dominating the pronoun.
2. Go up tree to first NP or S encountered.
   - Call node X, and path to it, p.
   - Search left-to-right below X and to left of p, proposing any NP node which has an NP or S between it and X.
3. If X is highest S node in sentence,
   - Search previous trees, in order of recency, left-to-right, breadth-first, proposing NPs encountered.
4. Otherwise, from X, go up to first NP or S node encountered,
   - Call this X, and path to it p.
5. If X is an NP, and p does not pass through an N-bar that X immediately dominates, propose X.
6. Search below X, to left of p, left-to-right, breadth-first, proposing NP encountered.
7. If X is an S, search below X to right of p, left-to-right, breadth-first, but not going through any NP or S, proposing NP encountered.
8. Go to 2.

Example:

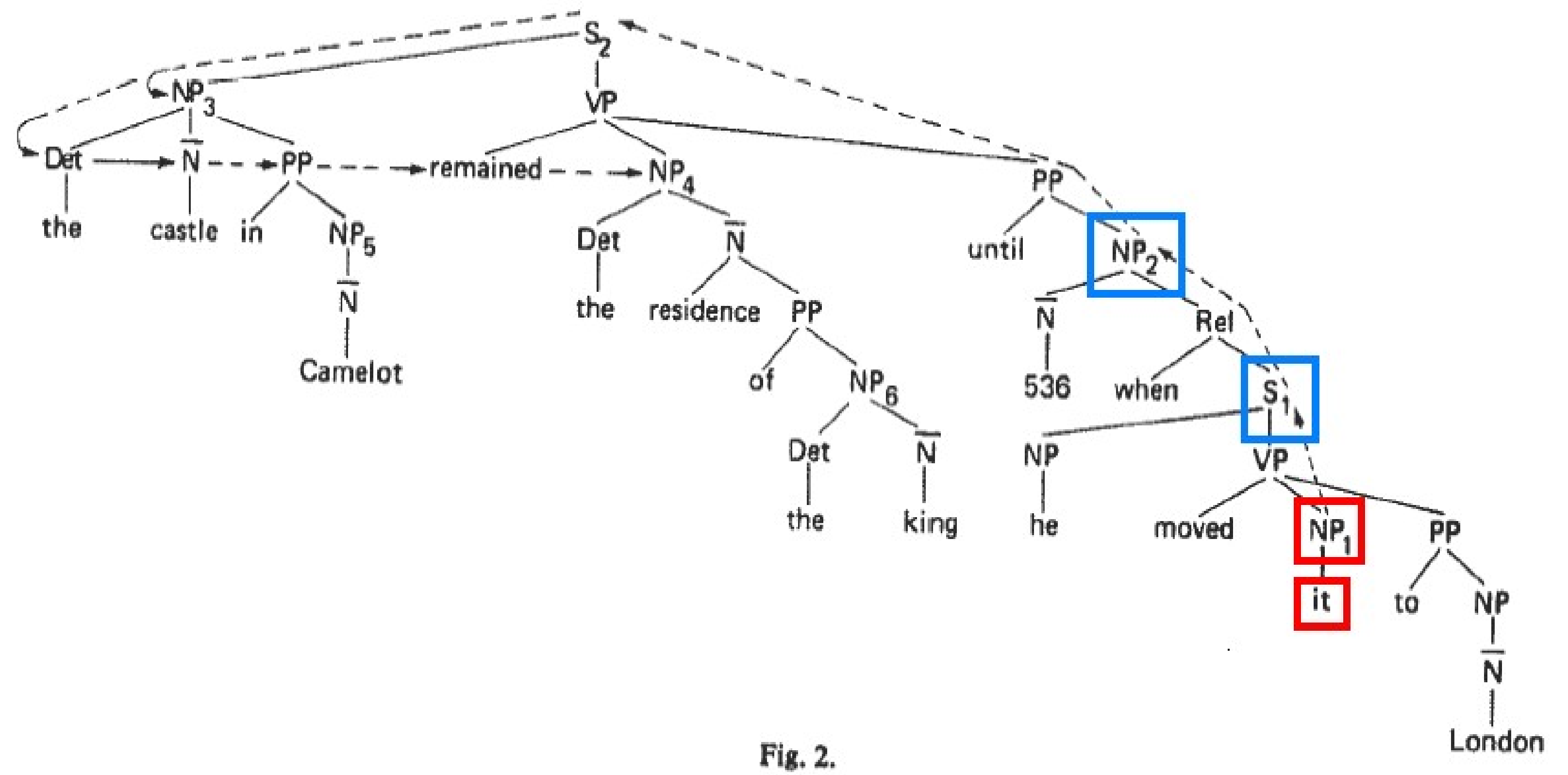


Fig. 2.

From X, go up to first NP or S node encountered. Call this X, and path to it p.

Hobbs’s “Naïve” Algorithm

1. Begin at the NP immediately dominating the pronoun.
2. Go up tree to first NP or S encountered.
   - Call node X, and path to it, p.
   - Search left-to-right below X and to left of p, proposing any NP node which has an NP or S between it and X.
3. If X is highest S node in sentence,
   - Search previous trees, in order of recency, left-to-right, breadth-first, proposing NPs encountered.
4. Otherwise, from X, go up to first NP or S node encountered,
   - Call this X, and path to it p.
5. If X is an NP, and p does not pass through an N-bar that X immediately dominates, propose X.
6. Search below X, to left of p, left-to-right, breadth-first, proposing NP encountered.
7. If X is an S, search below X to right of p, left-to-right, breadth-first, but not going through any NP or S, proposing NP encountered.
8. Go to 2.

Example:

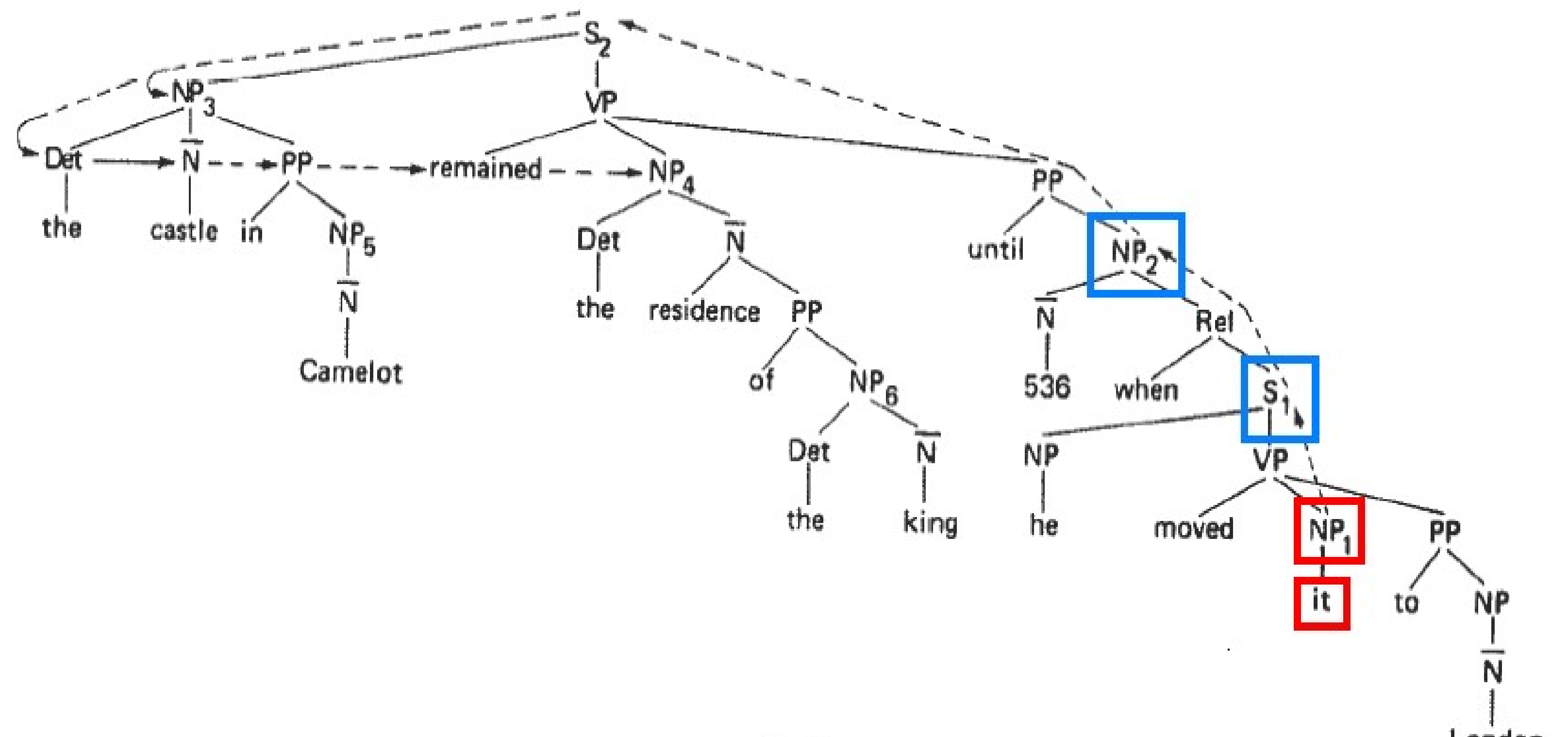


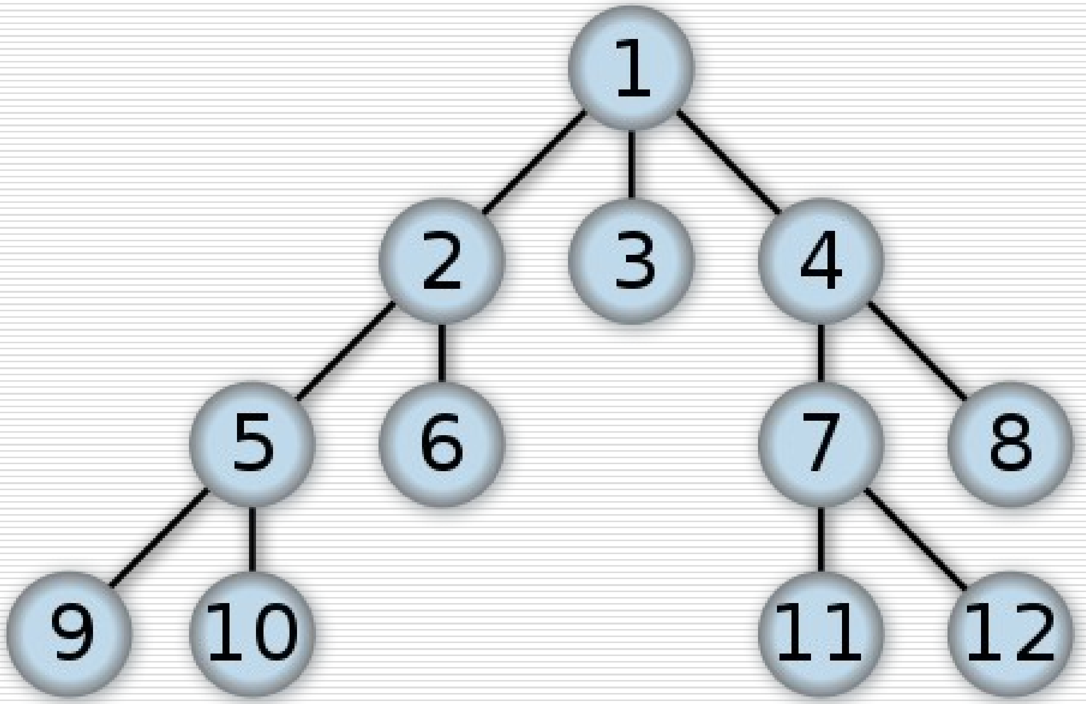
Fig. 2.

NP2 is proposed. Rejected by selectional restrictions (dates can't move).

Hobbs's "Naïve" Algorithm

1. Begin at the NP immediately dominating the pronoun.
2. Go up tree to first NP or S encountered.
   - Call node X, and path to it, p.
   - Search left-to-right below X and to left of p, proposing any NP node which has an NP or S between it and X.
3. If X is highest S node in sentence,
   - Search previous trees, in order of recency, left-to-right, breadth-first, proposing NPs encountered.
4. Otherwise, from X, go up to first NP or S node encountered,
   - Call this X, and path to it p.
5. If X is an NP, and p does not pass through an N-bar that X immediately dominates, propose X.
6. Search below X, to left of p, left-to-right, breadth-first, proposing NP encountered.
7. If X is an S, search below X to right of p, left-to-right, breadth-first, but not going through any NP or S, proposing NP encountered.
8. Go to 2.

Left-to-right, breadth-first search



Example:

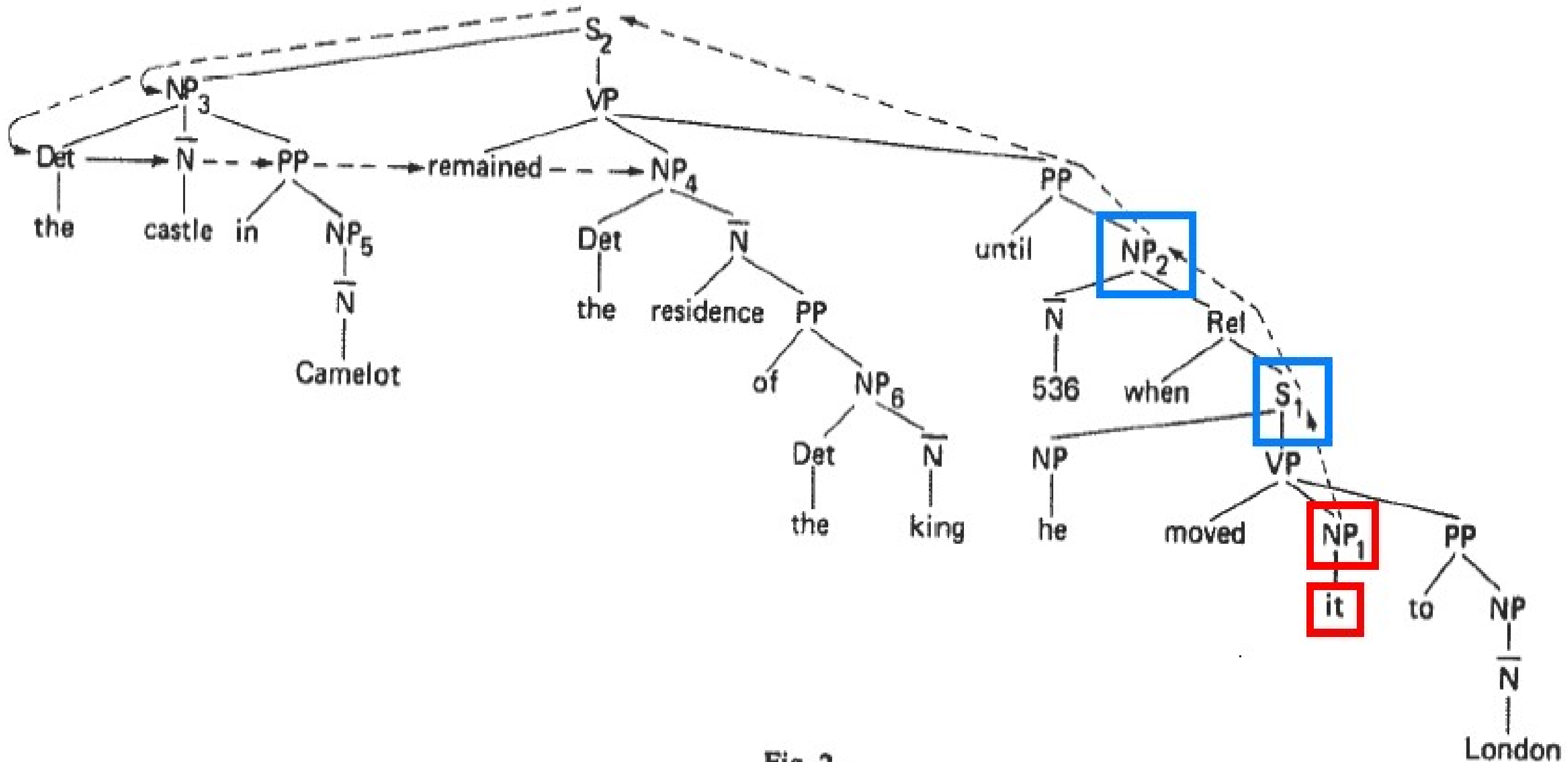


Fig. 2.

NP2: search yields no candidate. Go to next step of the algorithm.

Hobbs’s “Naïve” Algorithm

1. Begin at the NP immediately dominating the pronoun.
2. Go up tree to first NP or S encountered.
   - Call node X, and path to it, p.
   - Search left-to-right below X and to left of p, proposing any NP node which has an NP or S between it and X.
3. If X is highest S node in sentence,
   - Search previous trees, in order of recency, left-to-right, breadth-first, proposing NPs encountered.
4. Otherwise, from X, go up to first NP or S node encountered,
   - Call this X, and path to it p.
5. If X is an NP, and p does not pass through an N-bar that X immediately dominates, propose X.
6. Search below X, to left of p, left-to-right, breadth-first, proposing NP encountered.
7. If X is an S, search below X to right of p, left-to-right, breadth-first, but not going through any NP or S, proposing NP encountered.
8. Go to 2.

Hobbs’s “Naïve” Algorithm

1. Begin at the NP immediately dominating the pronoun.
2. Go up tree to first NP or S encountered.
   - Call node X, and path to it, p.
   - Search left-to-right below X and to left of p, proposing any NP node which has an NP or S between it and X.
3. If X is highest S node in sentence,
   - Search previous trees, in order of recency, left-to-right, breadth-first, proposing NPs encountered.
4. Otherwise, from X, go up to first NP or S node encountered,
   - Call this X, and path to it p.
5. If X is an NP, and p does not pass through an N-bar that X immediately dominates, propose X.
6. Search below X, to left of p, left-to-right, breadth-first, proposing NP encountered.
7. If X is an S, search below X to right of p, left-to-right, breadth-first, but not going through any NP or S, proposing NP encountered.
8. Go to 2.

Example:

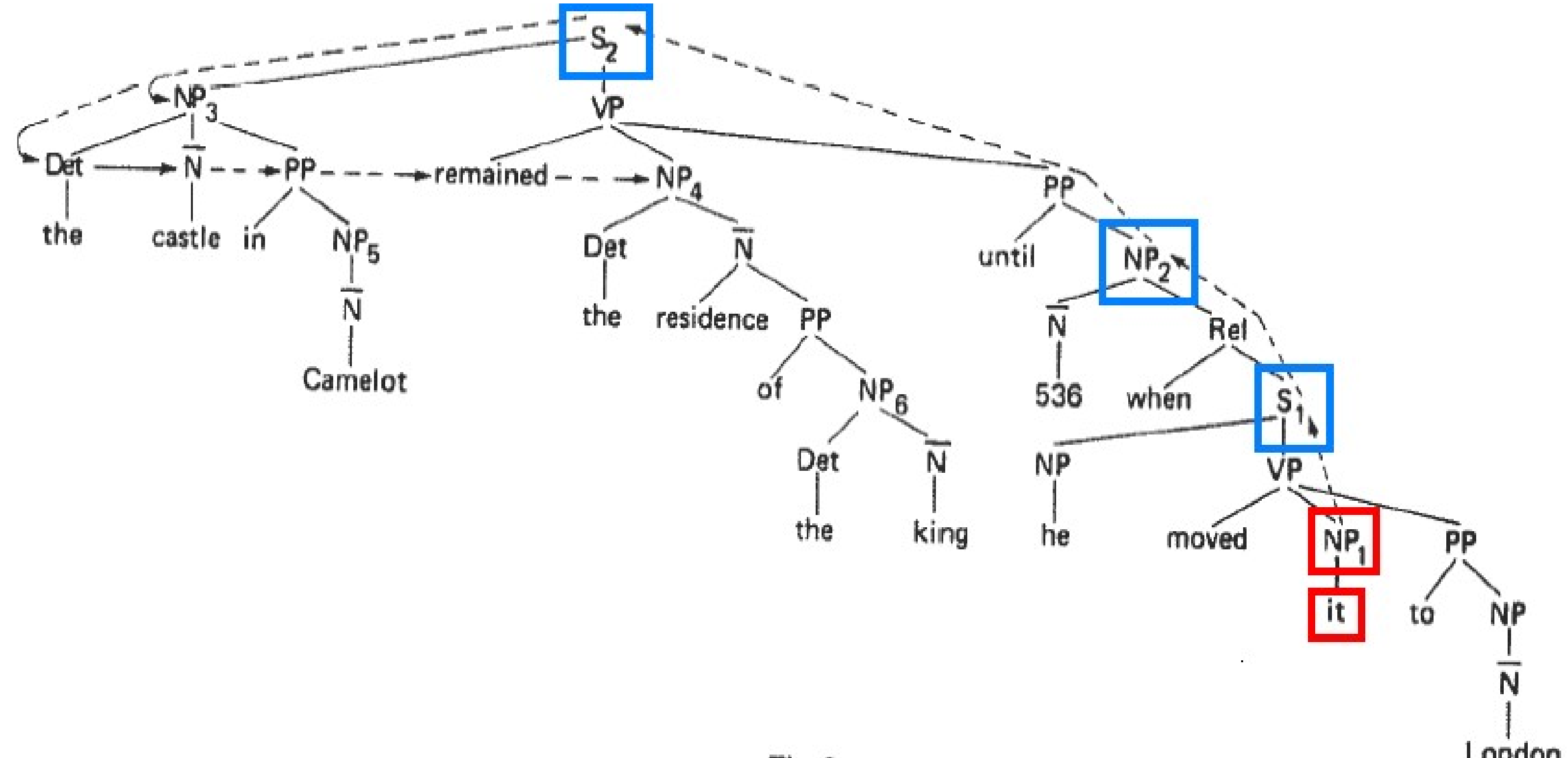


Fig. 2.

Search left-to-right below X and to left of p, proposing any NP node which has an NP or S between it and X.

Example:

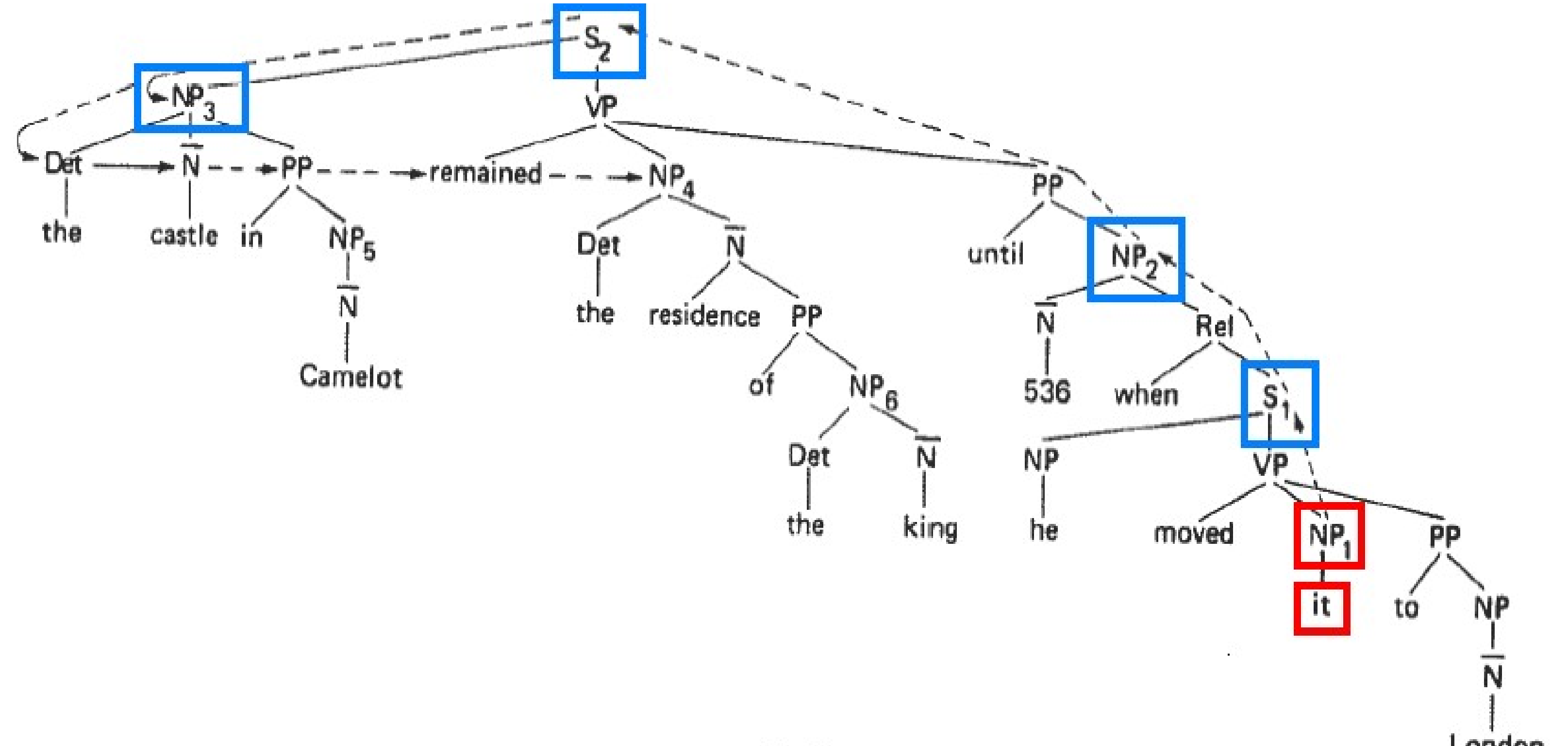


Fig. 2.

NP3: proposed. Rejected by rejected by selectional restrictions (can’t move large fixed objects.)

Example:

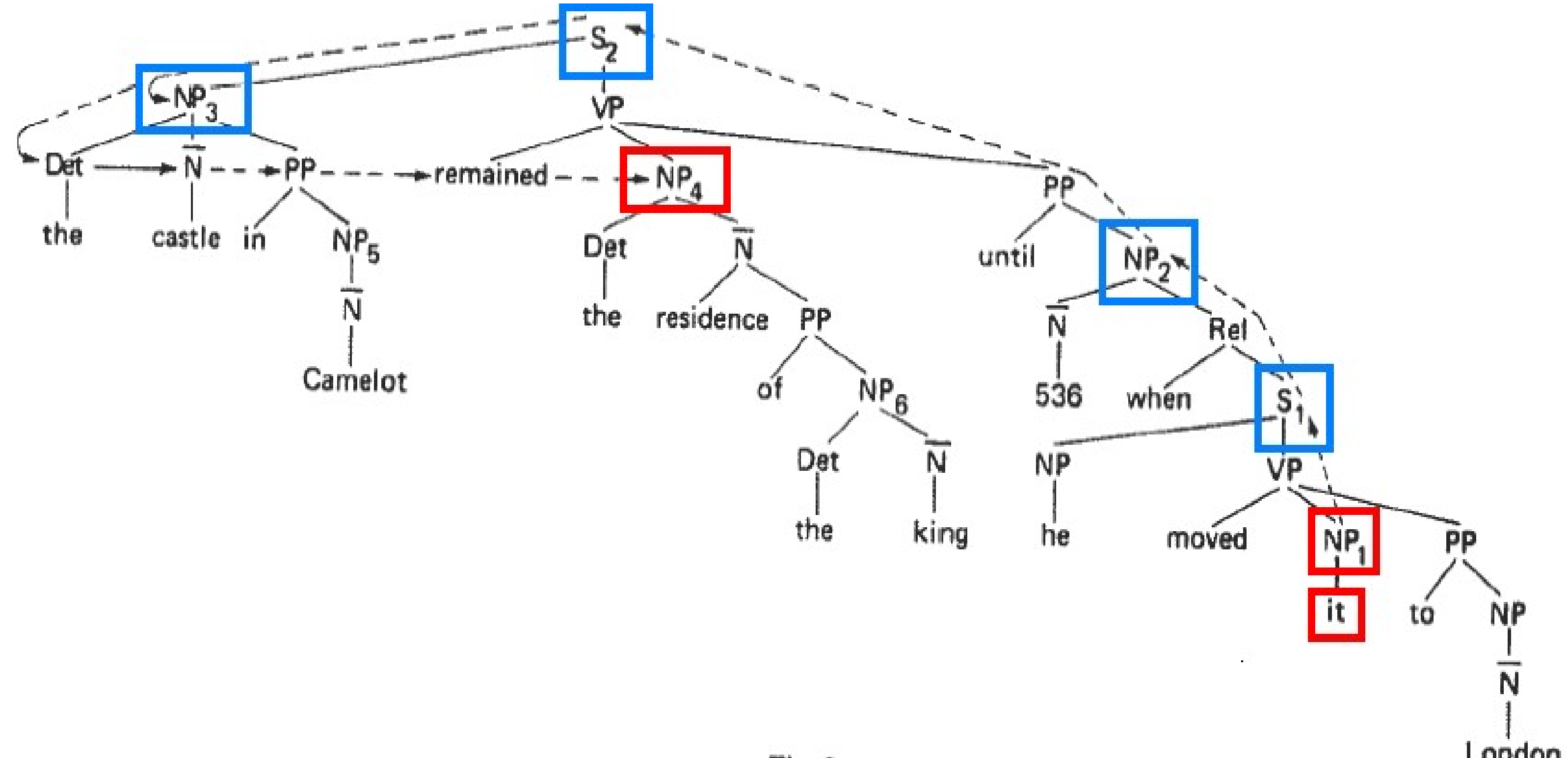


Fig. 2.

NP4: proposed. Accepted.

Another example:

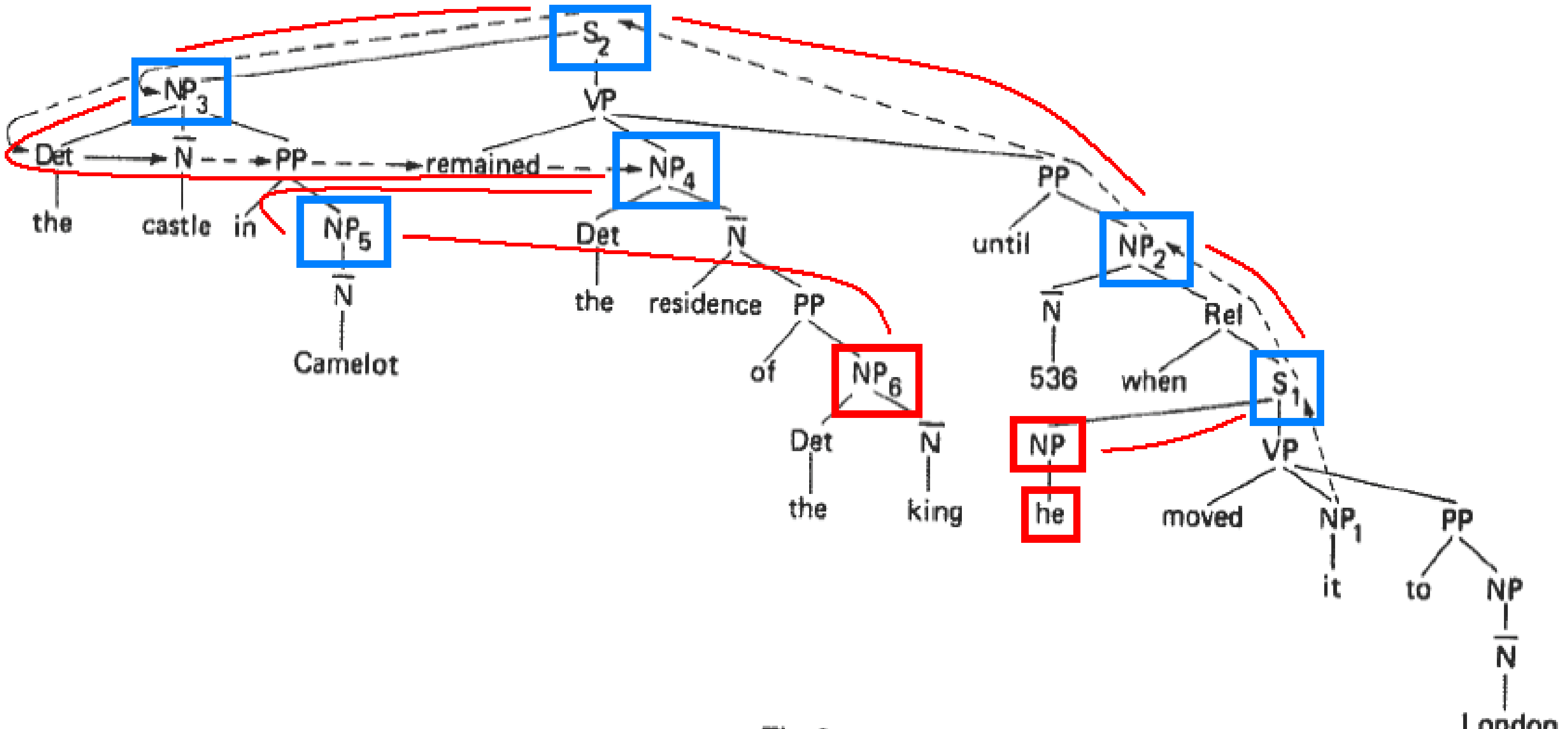


Fig. 2.

The referent for “he”: we follow the same path, get to the same place, but reject NP4, then reject NP5. Finally, accept NP6.

The algorithm: evaluation

- Corpus:
  - *Early civilization in China* (book, non-fiction)
  - *Wheels* (book, fiction)
  - *Newsweek* (magazine, non-fiction)

The algorithm: evaluation

- Hobbs analyzed, by hand, 100 consecutive examples from these three “very different” texts.

  - pronouns resolved: “he”, “she”, “it”, “they”
  - didn’t count “it” if it referred to a syntactically recoverable “that” clause – since, as he points out, the algorithm does just the wrong thing here.

- Assumed “the correct parse” was available.

The algorithm: results

- Overall, no selectional constraints: 88.3%
- Overall, with selectional constraints: 91.7%

The algorithm: results

- *This is somewhat deceptive since in over half the cases there was only one nearby plausible antecedent.* (p. 344)
- 132/300 times, there was a conflict
- 12/132 resolved by selectional constraints, 96/120 by algorithm

The algorithm: results

- Thus, 81.8% of the conflicts were resolved by a combination of the algorithm and selection.
- Without selectional restrictions, the algorithm was correct 72.7%.
- Hobbs concludes that the naïve approach provides a high baseline.
- Semantic algorithms will be necessary for much of the rest, but will not perform better for some time.

Centering Theory

- Grosz, Barbara J., Aravind Joshi, and Scott Weinstein. 1995. Centering: A framework for modeling the local coherence of discourse. *Computational Linguistics,* 21(2):203-225

- Brennan, Susan E., Marilyn W. Friedman, and Carl J. Pollard. 1987. A centering approach to pronouns. In *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics,* pages 155-162.

Centering Theory

Basic ideas:

- A discourse has a focus, or center.
- The center typically remains the same for a few sentences, then shifts to a new object.
- The center of a sentence is typically pronominalized.
- Once a center is established, there is a strong tendency for subsequent pronouns to continue to refer to it.

Some examples

Compare the two discourses:

- a. John went to his favorite music store to buy a piano.
- b. He had frequented the store for many years.
- c. He was excited that he could finally buy a piano.
- d. He arrived just as the store was closing for the day.

- a. John went to his favorite music store to buy a piano.
- b. It was a store John had frequented for many years.
- c. He was excited that he could finally buy a piano.
- d. It was closing just as John arrived.

Another example

- a. Terry really goofs sometimes.
- b. Yesterday was a beautiful day and he was excited about trying out his new sailboat.
- c. He wanted Tony to join him on a sailing expedition.
- d. He called him at 6 AM.
- e. He was sick and furious at being woken up so early.

Replace pronoun with proper name

- a. Terry really goofs sometimes.
- b. Yesterday was a beautiful day and he was excited about trying out his new sailboat.
- c. He wanted Tony to join him on a sailing expedition.
- d. He called him at 6 AM.
- e. Tony was sick and furious at being woken up so early.

We continue the story ...

- a. Terry really goofs sometimes.
- b. Yesterday was a beautiful day and he was excited about trying out his new sailboat.
- c. He wanted Tony to join him on a sailing expedition.
- d. He called him at 6 AM.
- e. Tony was sick and furious at being woken up so early.
- f. He told Terry to get lost and hung up.
- g. Of course, he hadn’t intended to upset Tony.

Once again, replace pronoun with proper name

- a. Terry really goofs sometimes.
- b. Yesterday was a beautiful day and he was excited about trying out his new sailboat.
- c. He wanted Tony to join him on a sailing expedition.
- d. He called him at 6 AM.
- e. Tony was sick and furious at being woken up so early.
- f. He told Terry to get lost and hung up.
- g. Of course, Terry hadn’t intended to upset Tony.

Another example

- Compare the two discourses

  - 1 a. John was very worried last night.
  - b. He called Bob.
  - c. He told him that there was a big problem.

  - 2 a. John was very worried last night.
  - b. He called Bob.
  - c. He told him never to call again at such a late hour.

Again, replace pronoun with proper name

- Compare the two discourses
  - 1 a. John was very worried last night.
  - b. He called Bob.
  - c. He told him that there was a big problem.
  - 2 a. John was very worried last night.
  - b. He called Bob.
  - c. Bob told him never to call again at such a late hour.

When are pronouns better than proper names?

- a. Susan gave Betsy a pet hamster.
- b. She reminded her that such hamsters are quite shy.

Compare the following alternative utterances.

- c1. She asked Betsy whether she liked the gift.
- c2. Susan asked her whether she liked the gift.

- c3. Betsy told her that she really liked the gift.
- c4. She told Susan that she really liked the gift.

Centering

- Centering theory was developed by Barbara J. Grosz, Aravind K. Joshi and Scott Weinstein in the 1980s to explain this kind of phenomena.

Definitions

Utterance – A sentence in the context of a discourse.

Center – An entity referred to in the discourse (our discourse referents).

Forward looking centers – An utterance U_n is assigned a set of centers $C_f(U_n)$ that are referred to in U_n (basically, the drefs introduced / acccessed in a sentence).

Backward looking center – An utterance U_n is assigned a single center $C_b(U_n)$, which is equal to one of the centers in $C_f(U_{n-1}) \cap C_f(U_n)$.

If there is no such center, $C_b(U_n)$ is NIL.

Ranking of forward looking centers

- $C_f(U_n)$ is an ordered set.
- Its order reflects the prominence of the centers in the utterance.
- The ordering (ranking) is done primarily according to the syntactic position of the word in the utterance (subject > object(s) > other).
- The prominent center of an utterance, $C_p(U_n)$, is the highest ranking center in $C_f(U_n)$.

Ranking of forward looking centers

- Think of the backward looking center $C_b(U_n)$ as the current topic.

- Think of the preferred center $C_p(U_n)$ as the potential new topic.

Constraints on centering

1. There is precisely one C_b.

1. Every element of $C_f(U_n)$ must be realized in U_n.

1. $C_b(U_n)$ is the highest-ranked element of $C_f(U_{n-1})$ that is realized in U_n.

Another example

- U_1. John drives a Ferrari.
- U_2. He drives too fast.
- U_3. Mike races him often.
- U_4. He sometimes beats him.

Let’s see what the centers are…

- U_1. John drives a Ferrari.
  $C_b(U_1)$ = NIL (or: John). $C_f(U_1)$ = (John, Ferrari)

- U_2. He drives too fast.
  $C_b(U_2)$ = John. $C_f(U_2)$ = (John)

- U_3. Mike races him often.
  $C_b(U_3)$ = John. $C_f(U_3)$ = (Mike, John)

- U_4. He sometimes beats him.
  $C_b(U_4)$ = Mike. $C_f(U_4)$ = (Mike, John)

Types of transitions

Transition Type from U_{n-1} to U_n	$C_b(U_n) = C_b(U_{n-1})$	$C_b(U_n) = C_p(U_n)$
Center Continuation	+	+
Center Retaining	+	-
Center Shifting-1	-	+
Center Shifting	-	-

Let’s see what the transitions are…

- U_1. John drives a Ferrari.
  $C_b(U_1)$ = John. $C_f(U_1)$ = (John, Ferrari)

- U_2. He drives too fast.
  $C_b(U_2)$ = John. $C_f(U_2)$ = (John)

Types of transitions

Transition Type from U_{n-1} to U_n	$C_b(U_n) = C_b(U_{n-1})$	$C_b(U_n) = C_p(U_n)$
Center Continuation	+	+
Center Retaining	+	-
Center Shifting-1	-	+
Center Shifting	-	-

Let's see what the transitions are…

- U_1. John drives a Ferrari.
  $C_b(U_1)$ = John. $C_f(U_1)$ = (John, Ferrari)

- U_2. He drives too fast. (**continuation**)
  $C_b(U_2)$ = John. $C_f(U_2)$ = (John)

Let's see what the transitions are…

- U_1. John drives a Ferrari.
  $C_b(U_1)$ = John. $C_f(U_1)$ = (John, Ferrari)

- U_2. He drives too fast. (**continuation**)
  $C_b(U_2)$ = John. $C_f(U_2)$ = (John)

- U_3. Mike races him often.
  $C_b(U_3)$ = John. $C_f(U_3)$ = (Mike, John)

Types of transitions

Transition Type from U_{n-1} to U_n	$C_b(U_n) = C_b(U_{n-1})$	$C_b(U_n) = C_p(U_n)$
Center Continuation	**+**	**+**
Center Retaining	**+**	**-**
Center Shifting-1	**-**	**+**
Center Shifting	**-**	**-**

Let’s see what the transitions are…

- U_1. John drives a Ferrari.
  $C_b(U_1)$ = John. $C_f(U_1)$ = (John, Ferrari)

- U_2. He drives too fast. (**continuation**)
  $C_b(U_2)$ = John. $C_f(U_2)$ = (John)

- U_3. Mike races him often. (**retaining**)
  $C_b(U_3)$ = John. $C_f(U_3)$ = (Mike, John)

Let's see what the transitions are…

- U_1. John drives a Ferrari.
  $C_b(U_1)$ = John. $C_f(U_1)$ = (John, Ferrari)

- U_2. He drives too fast. (**continuation**)
  $C_b(U_2)$ = John. $C_f(U_2)$ = (John)

- U_3. Mike races him often. (**retaining**)
  $C_b(U_3)$ = John. $C_f(U_3)$ = (Mike, John)

- U_4. He sometimes beats him.
  $C_b(U_4)$ = Mike. $C_f(U_4)$ = (Mike, John)

Types of transitions

Transition Type from U_{n-1} to U_n	$C_b(U_n) = C_b(U_{n-1})$	$C_b(U_n) = C_p(U_n)$
Center Continuation	+	+
Center Retaining	+	-
Center Shifting-1	-	+
Center Shifting	-	-

Let's see what the transitions are...

- U_1. John drives a Ferrari.
  $C_b(U_1)$ = John. $C_f(U_1)$ = (John, Ferrari)

- U_2. He drives too fast. (**continuation**)
  $C_b(U_2)$ = John. $C_f(U_2)$ = (John)

- U_3. Mike races him often. (**retaining**)
  $C_b(U_3)$ = John. $C_f(U_3)$ = (Mike, John)

- U_4. He sometimes beats him. (**shifting-1**)
  $C_b(U_4)$ = Mike. $C_f(U_4)$ = (Mike, John)

Centering rules in discourse

1. If some element of $C_f(U_{n-1})$ is realized as a pronoun in U_n, then so is $C_b(U_n)$.

1. Continuation is preferred over retaining, which is preferred over shifting-1, which is preferred over shifting:

   Cont >> Retain >> Shift-1 >> Shift

Violation of rule 1

- Assuming *He* in utterance U_1 refers to *John*…

- U_1. He has been acting quite odd.
- U_2. He called up Mike Yesterday.
- U_3. John wanted to meet him urgently.

In more detail ...

- U_1. He has been acting quite odd.
  $C_b(U_1)$ = John. $C_f(U_1)$ = (John)

- U_2. He called up Mike Yesterday.
  $C_b(U_2)$ = John. $C_f(U_2)$ = (John, Mike)

- U_3. John wanted to meet him urgently.
  $C_b(U_3)$ = John. $C_f(U_3)$ = (John, Mike)

Violation of rule 2

Compare the two discourses we started with:

- U_1. John went to his favorite music store to buy a piano.
- U_2. He had frequented the store for many years.
- U_3. He was excited that he could finally buy a piano.
- U_4. He arrived just as the store was closing for the day.

- U_1. John went to his favorite music store to buy a piano.
- U_2. It was a store John had frequented for many years.
- U_3. He was excited that he could finally buy a piano.
- U_4. It was closing just as John arrived.

Transitions for the 1st discourse

- U_1. John went to his favorite music store to buy a piano.
  $C_b(U_1)$ = John. $C_f(U_1)$ = (John, store, piano).
- U_2. He had frequented the store for many years.
  $C_b(U_2)$ = John. $C_f(U_2)$ = (John, store). **CONT**
- U_3. He was excited that he could finally buy a piano.
  $C_b(U_3)$ = John. $C_f(U_3)$ = (John, piano). **CONT**
- U_4. He arrived just as the store was closing for the day.
  $C_b(U_4)$ = John. $C_f(U_4)$ = (John, store). **CONT**

Transitions for the 2nd discourse

- U_1. John went to his favorite music store to buy a piano. $C_b(U_1)$ = John. $C_f(U_1)$ = (John, store, piano).
- U_2. It was a store John had frequented for many years. $C_b(U_2)$ = John. $C_f(U_2)$ = (store, John). **RETAIN**
- U_3. He was excited that he could finally buy a piano. $C_b(U_3)$ = John. $C_f(U_3)$ = (John, piano). **CONT**
- U_4. It was closing just as John arrived. $C_b(U_4)$ = John. $C_f(U_4)$ = (store, John). **RETAIN**

Centering algorithm

- An algorithm for centering and pronoun binding has been presented by Susan E. Brennan, Marilyn W. Friedman and Carl J. Pollard, based on the centering theory we have just discussed.

General structure of algorithm

For each utterance perform the following steps

Anchor construction: Create all possible anchors (pairs of forward centers and a backward center).

Anchor filtering: Filter out the bad anchors according to various filters.

Anchor ranking: Rank the remaining anchors according to their transition type.

General structure of algorithm

This is very similar to the general architecture of the algorithm in Lappin & Leass 1994:

- First: filtering based on hard constraints
- Then: ranking based on some soft constraints

Construction of the anchors

1. Create a list of referring expressions (REs) in the utterance, ordered by grammatical relation.

1. Expand each RE into a center according to whether it is a pronoun or a proper name. In case of pronouns, the agreement features must match.

1. Create a set of backward centers according to the forward centers of the previous utterance, plus NIL.

1. Create a set of anchors, which is the Cartesian product of the possible backward and forward centers.

Filtering the proposed anchors

- The constructed anchors undergo the following filters.

1. Remove all anchors that assign the same center to two syntactic positions that cannot co-index (binding theory).

1. Remove all anchors which violate constraint 3, i.e. whose C_b is not the highest ranking center of the previous C_f which appears in the anchor's C_f list.

1. Remove all anchors which violate rule 1. If the utterance has pronouns then remove all anchors where the C_b is not realized by a pronoun.

Ranking the anchors

- Classify, every anchor that passed the filters, into its transition type (cont, retain, shift-1, shift).
- Choose the anchor with the most preferable transition type according to rule 2.

Let's look at an example

- U_1. John likes to drive fast.
  $C_b(U_1)$ = John. $C_f(U_1)$ = (John)

- U_2. He races Mike.
  $C_b(U_2)$ = John. $C_f(U_2)$ = (John, Mike)

- U_3. Mike beats him sometimes.

Let's generate the anchors for $U_{3.}$

Anchor construction for U_3

- $C_b(U_2)$ = John. $C_f(U_2)$ = (John, Mike)
- U_3. Mike beats him sometimes.

1. Create a list of REs.

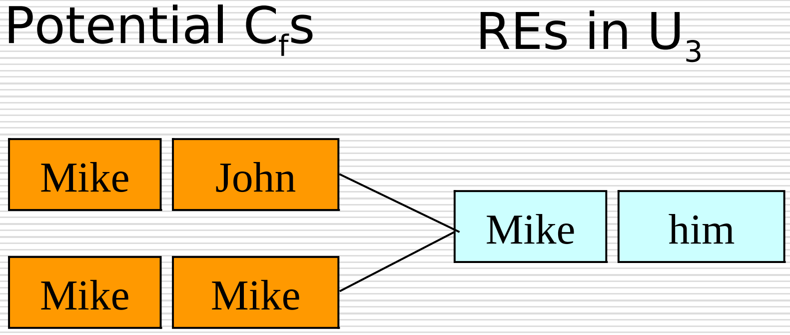
REs in U_3

Mike	him

Anchor construction for U_3

- $C_b(U_2)$ = John. $C_f(U_2)$ = (John, Mike)
- U_3. Mike beats him sometimes.

1. Create a list of REs.
2. Expand into possible forward center lists.

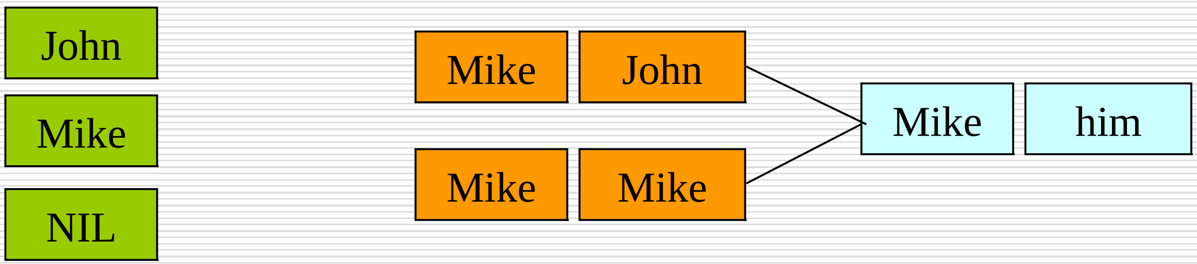


Anchor construction for U_3

- $C_b(U_2)$ = John. $C_f(U_2)$ = (John, Mike)
- U_3. Mike beats him sometimes.

1. Create a list of REs.
2. Expand into possible forward center lists.
3. Create possible backward centers according to $C_f(U_2)$.

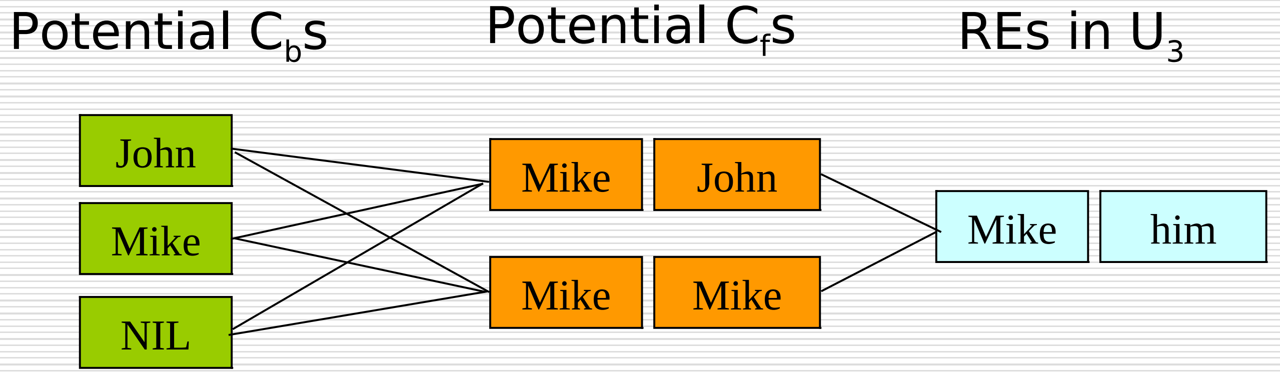
Potential C_bs Potential C_fs REs in U_3



Anchor construction for U_3

- $C_b(U_2)$ = John. $C_f(U_2)$ = (John, Mike)
- U_3. Mike beats him sometimes.

1. Create a list of REs.
2. Expand into possible forward center lists.
3. Create possible backward centers according to $C_f(U_2)$.
4. Create a list of all anchors (cartesian product).



Anchor construction for U_3

- $C_b(U_2)$ = John. $C_f(U_2)$ = (John, Mike)
- U_3. Mike beats him sometimes.

1. Create a list of REs.
2. Expand into possible forward center lists.
3. Create possible backward centers according to $C_f(U_2)$.
4. Create a list of all anchors (cartesian product).

C_b	John	John	Mike	Mike	NIL	NIL
Mike	Mike	Mike	Mike	Mike	Mike	Mike
him	John	Mike	John	Mike	John	Mike

Filtering the anchors

- $C_b(U_2)$ = John. $C_f(U_2)$ = (John, Mike)
- U_3. Mike beats him sometimes.

1. Remove all anchors that assign the same center to two syntactic positions that cannot co-index.

C_b	John	John	Mike	Mike	NIL	NIL
Mike	Mike	Mike	Mike	Mike	Mike	Mike
him	John	Mike	John	Mike	John	Mike

Filtering the anchors

- $C_b(U_2)$ = John. $C_f(U_2)$ = (John, Mike)
- U_3. Mike beats him sometimes.

1. Remove all anchors that assign the same center to two syntactic positions that cannot co-index.
2. Remove all anchors which violate constraint 3, i.e. whose C_b is not the highest ranking center in $C_f(U_2)$ which appears in the anchor's C_f.

C_b	John	John	Mike	Mike	NIL	NIL
Mike	Mike	Mike	Mike	Mike	Mike	Mike
him	John	Mike	John	Mike	John	Mike

Filtering the anchors

- $C_b(U_2)$ = John. $C_f(U_2)$ = (John, Mike)
- U_3. Mike beats him sometimes.

1. Remove all anchors that assign the same center to two syntactic positions that cannot co-index.
2. Remove all anchors which violate constraint 3, i.e. whose C_b is not the highest ranking center in $C_f(U_2)$ which appears in the anchor's C_f.
3. Remove all anchors which violate rule 1, i.e. the C_b must be realized by a pronoun.

C_b	John	John	Mike	Mike	NIL	NIL
Mike	Mike	Mike	Mike	Mike	Mike	Mike
him	John	Mike	John	Mike	John	Mike

Ranking the anchors

- $C_b(U_2)$ = John. $C_f(U_2)$ = (John, Mike)
- The only remaining anchor –
  $C_b(U_3)$ = John. $C_f(U_3)$ = (Mike, John)
- RETAIN

Evaluation

- The algorithm waits until the end of a sentence to resolve references, whereas humans appear to do this on-line.

Evaluation (example from Kehler)

- a. Terry really gets angry sometimes.
- b. Yesterday was a beautiful day and he was excited about trying out his new sailboat.
- c. He wanted Tony to join him on a sailing expedition, and left him a message on his answering machine. [C_b=C_p=Terry]
- d. Tony called him at 6AM the next morning. [C_b=Terry, C_p=Tony]

- e1. He was furious for being woken up so early.
- e2. He was furious with him for being woken up so early.
- e3. He was furious with Tony for being woken up so early.

Evaluation (example from Kehler)

[C_b=Terry, C_p=Tony]

- e1. He was furious for being woken up so early.
- e2. He was furious with him for being woken up so early.
- e3. He was furious with Tony for being woken up so early.

BFP algorithm picks:

- – Terry for e1 (preferring CONT over SHIFT-1)
- – Tony for e2 (preferring SHIFT-1 over SHIFT)
- – ?? for e3, as this violates Constraint 1, but would be a SHIFT otherwise

Evaluation (example from Kehler)

- Reference seems to be influenced by the rhetorical relations between sentences, which BFP is not sensitive to.

Centering vs Hobbs

- Marilyn A. Walker. 1989, Evaluating discourse processing algorithms. In Proceedings of ACL 27, Vancouver, British Columbia, 251-261.

- Walker 1989 manually compared a version of centering to Hobbs on 281 examples from three genres of text.

- Reported 81.8% for Hobbs, 77.6% centering.

Corpus-based Evaluation of Centering

- Massimo Poesio, Rosemary Stevenson, Barbara Di Eugenio, Janet Hitzeman. 2004. Centering: A Parametric Theory and Its Instantiations. Computational Linguistics 30:3, 309 – 363

Comparison

- “A long-standing weakness in the area of anaphora resolution: the inability to fairly and consistently compare anaphora resolution algorithms due not only to the difference of evaluation data used, but also to the diversity of pre-processing tools employed by each system.” (Barbu & Mitkov, 2001)

- It’s customary to evaluate algorithms on the MUC-6 and MUC-7 coreference corpora.
  - http://www.cs.nyu.edu/cs/faculty/grishman/muc6.html
  - http://www.itl.nist.gov/iaui/894.02/related_projects/muc/proceedings/muc_7_toc.html
  - http://www.aclweb.org/anthology-new/M/M98/M98-1029.pdf