Pragmatics

- Study of the relation between language and its context of use
- Includes the study of how language can be used to refer to people and things
- Includes the study of the discourse structure

Discourse

Gracie: Oh yeah... And then Mr. and Mrs. Jones were having matrimonial trouble, and my brother was hired to watch Mrs. Jones.
George: Well, I imagine she was a very attractive woman.
Gracie: She was, and my brother watched her day and night for six months.
George: Well, what happened?
Gracie: She finally got a divorce.
George: Mrs. Jones?
Gracie: No, my brother's wife.

George Burns and Gracie Allen in *The Salesgirl*

Discourse

- Language consists of related, collocated group of sentences. We refer to such a group of sentences as a **discourse**.
- Two basic forms of discourse:
  1. Monologue
  2. Dialogue

Reference Resolution

- **Reference**: the process by which speakers use expressions to denote an entity.
- **Referring expression**: expression used to perform reference.
- **Example**: Raju went home. He forgot to take his bag.

Referent: the entity that is referred to.
Co reference: referring expressions that are used to refer to the same entity.
Anaphora: reference to a previously introduced entity.

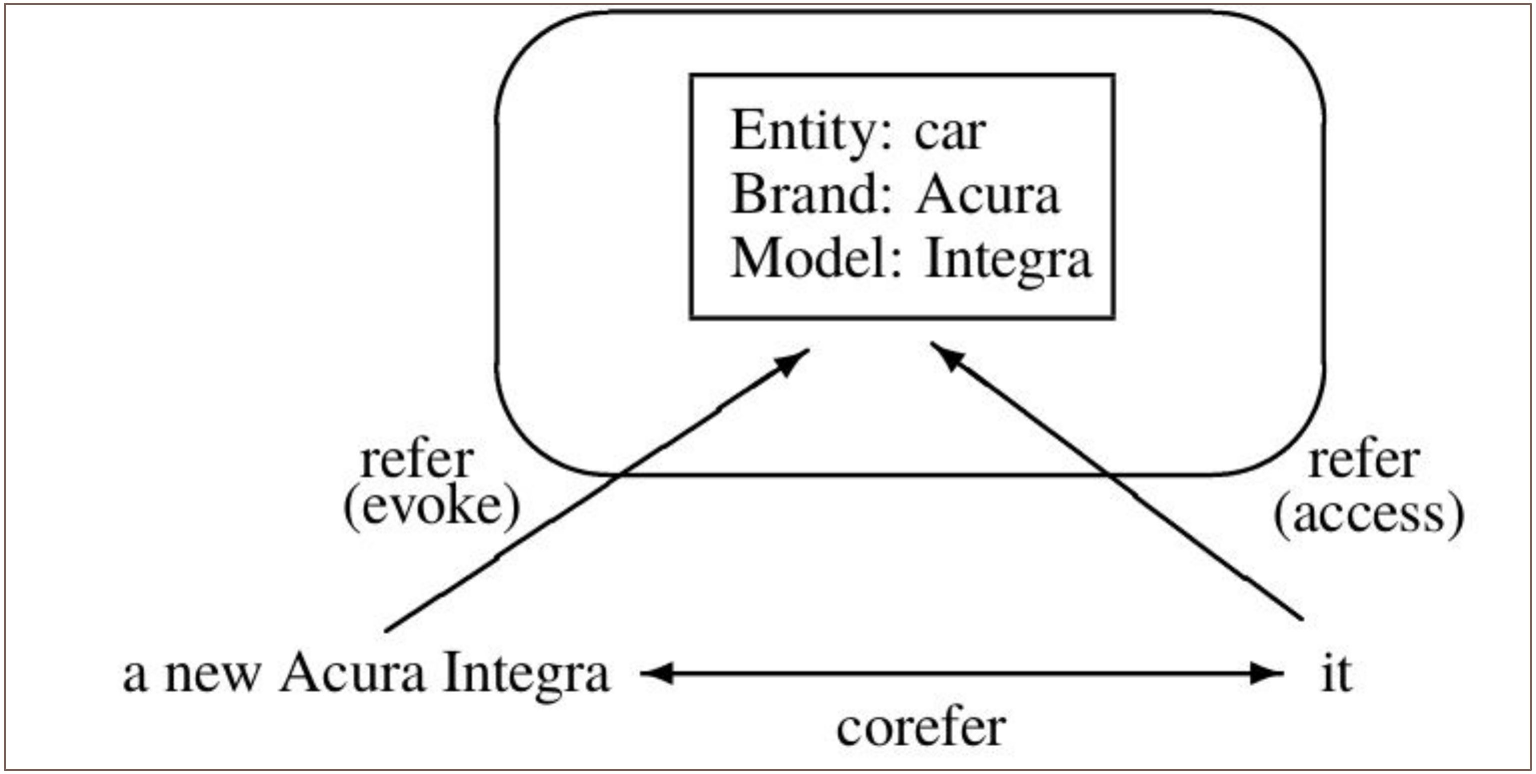
- previously introduced entity is **antecedent**
- referring expression is said to be **anaphoric**.

Cataphora: reference are mentioned before their referents

Discourse Model

- It contains representations of the entities that have been referred to in the discourse and the relationships in which they participate.
- Two components required by a system to produce and interpret referring expressions.
  - A method for constructing a discourse model that evolves dynamically.
  - A method for mapping between referring expressions and referents.

Discourse Model



John went to Bill's car dealership to check out an Acura Integra.
He looked at it for about an hour.

Reference Phenomena

Five common types of referring expression	
Type	**Example**
Indefinite noun phrase	I saw **a Ford Escort** today. {a,an,some}
Definite noun phrase	I saw a Ford Escort today. **The Escort** was white.
Pronoun	I saw a Ford Escort today. **It** was white.
Demonstratives	I like **this** better than **that**.
One-anaphora	I saw 6 Ford Escorts today. Now I want **one**.

Reference Phenomena

Three types of referents that complicate the reference resolution	
Type	**Example**
Inferrables	I almost bought a Ford Escort, but a **door** had a dent.
Discontinuous Sets	**John** has an Acura, and **Mary** has a Mazda. **They** often drive **them**.
Generics	I saw 6 Ford Escorts today. **They** are the coolest cars.

Reference Resolution

- How to develop successful algorithms for reference resolution? There are two necessary steps.
- First is to filter the set of possible referents by certain hard-and-fast constraints
- Second is to set the preference for possible referents

Constraints (for English)

- **Number Agreement**:
  - To distinguish between singular and plural references.
    - *John has a new car. <u>They</u> are red.
    - John has a new car. It is red.
- **Gender Agreement**:
  - To distinguish male, female, and non-personal genders.
    - John has a new car. He is attractive. [He = John]
    - John has a new car. It is attractive.[It = new car]
- **Person and Case Agreement**:
  - To distinguish between three forms of person;
    - *You and I have Escorts. <u>They</u> love them.
    - You and I have Escorts. <u>We</u> love them.
  - To distinguish between subject position, object position, and genitive position.

Constraints (for English)

- **Syntactic Constraints**:
  - Syntactic relationships between a referring expression and a possible antecedent noun phrase
    - John bought himself a new car. [himself=John]
    - John bought him a new car. [him≠John]
- **Selectional Restrictions**:
  - A verb places restrictions on its arguments.
    - John parked his Acura in the garage. He had driven it around for hours. [it=Acura, it≠garage];
    - I picked up the book and sat in a chair. It broke.[It=chair]

Preferences in Pronoun Interpretation

- **Recency**:
  - Entities introduced recently are more salient than those introduced before.
    - John has a Legend. Bill has an Escort. Mary likes to drive <u>it</u>.
- **Grammatical Role**:
  - Entities mentioned in subject position are more salient than those in object position.
    - Bill went to the Acura dealership with John. <u>He</u> bought an Escort. [he=Bill]

Preferences in Pronoun Interpretation

- **Repeated Mention**:
  - Entities that have been focused on in the prior discourse are more salient.

John needed a car to get to his new job.
He decided that he wanted something sporty.
Bill went to the Acura dealership with him.
He bought an Integra. [he=John]

- **Parallelism**
  - strong preferences appear to be induced by parallelism

Mary went with Sue to the Acura dealership. Sally went with her to the Mazda dealership. [her = Sue]

Preferences in Pronoun Interpretation

- **Verb Semantics**:
  - Certain verbs appear to place a semantically-oriented emphasis on one of their argument positions.
    - John telephoned Bill. He had lost the book in the mall. [He = John]
    - John criticized Bill. He had lost the book in the mall. [He = Bill]
    - David praised Hans because he … [he = Hans]
    - David apologized to Hans because he… [he = David]

Preferences in Pronoun Interpretation

- **World knowledge in general**:
  - The city council denied the demonstrators a permit because they {feared|advocated} violence.
  - **The city council** denied the demonstrators a permit because **they** {**feared**|advocated} violence.
  - The city council denied **the demonstrators** a permit because **they** {feared|**advocated**} violence.

Lappin and Leass 1994

- Idea: Maintain a discourse model , in which there are representations for potential referents.

- Lappin and Leass 1994 propose a discourse model in which potential referents have degrees of salience.

- They try to resolve (pronoun) references by finding highly salient referents compatible with pronoun agreement features.

- In effect, they incorporate:
  - recency
  - syntax-based preferences
  - agreement, but no (other) semantics

Lappin and Leass 1994

- First, we assign a number of salience factors & salience values to each referring expression.

- The salience values (weights) are arrived by experimentation on a certain corpus.

Lappin and Leass 1994

Salience Factor	Salience Value
Sentence recency	100
Subject emphasis	80
Existential emphasis	70
Accusative emphasis	50
Indirect object emphasis	40
Non-adverbial emphasis	50
Head noun emphasis	80

Example of these positions (shown in italics)

(18.62) *An Acura Integra* is parked in the lot. (subject)

(18.63) There is *an Acura Integra* parked in the lot. (existential predicate nominal)

(18.64) John parked *an Acura Integra* in the lot. (object)

(18.65) John gave *his Acura Integra* a bath. (indirect object)

(18.66) Inside *his Acura Integra*, John showed Susan his new CD player. (demarcated adverbial PP)

(18.67) The owner's manual for *an Acura Integra* is on John's desk.

Lappin and Leass 1994

- The algorithm employs a simple weighting scheme that integrates the effects of several preferences:
  - For each new entity, a representation for it is added to the discourse model and **salience value** computed for it.
  - Salience value is computed as the sum of the weights assigned by a set of **salience factors**.
    - The weight a salience factor assigns to a referent is the highest one the factor assigns to the referent's referring expression.
  - Salience values are cut in half each time a new sentence is processed.

Lappin and Leass 1994

The steps taken to resolve a pronoun are as follows:

- Collect potential referents (four sentences back);
- Remove potential referents that don’t semantically agree;
- Remove potential referents that don’t syntactically agree;
- Compute salience values for the rest potential referents;
- Select the referent with the highest salience value.

Example (from Jurafsky and Martin)

- John saw a beautiful Acura Integra at the dealership.
- He showed it to Bob.
- He bought it.

Example

- John saw a beautiful Acura Integra at the dealership.

Referent	Phrases	Value
John	{John}	?
Integra	{a beautiful Acura Integra}	?
dealership	{the dealership}	?

	Rec	Subj	Exist	Obj	Ind-Obj	Non-Adv	Head N	Total
John	100	80				50	80	310
Integra	100			50		50	80	280
dealership	100					50	80	230

Example

- He showed it to Bob.

Reduced by half

Referent	Phrases	Value
John	{ *John* }	155
Integra	{ *a beautiful Acura Integra* }	140
dealership	{ *the dealership* }	115

The first noun phrase in the second sentence is the pronoun **he**. Because he specifies male gender, John is the only possible referent

Example

- He showed it to Bob.

The pronoun ***he*** is in the current sentence (recency=100), subject position (=80), not in an adverbial (=50), and not embedded (=80), and so a total of 310 is added to the current weight for John:

Referent	Phrases	Value
John	{ *John*, he_1 }	465
Integra	{ *a beautiful Acura Integra* }	140
dealership	{ *the dealership* }	115

Since "Integra" is more salient than "dealership" (140>115):

- "it" refers to "Integra"

Example

- He showed it to Bob.

  Since ***it*** is in a non-embedded object position, **it** receives a weight of 100+50+50+80=280, and is added to the current weight for the Integra.

Referent	Phrases	Value
John	{ *John*, he_1 }	465
Integra	{ *a beautiful Acura Integra*, it_1 }	420
dealership	{ *the dealership* }	115

Example

- He showed it to Bob.

  Since **Bob** occupies an oblique argument position, it receives a weight of 100+40+50+80=270.

Referent	Phrases	Value
John	{ *John*, he_1 }	465
Integra	{ *a beautiful Acura Integra*, it_1 }	420
Bob	{ *Bob* }	270
dealership	{ *the dealership* }	115

Example

- He bought it.

Referent	Phrases	Value
John	{ *John*, he_1 }	232.5
Integra	{ *a beautiful Acura Integra*, it_1 }	210
Bob	{ *Bob* }	135
dealership	{ *the dealership* }	57.5

He = ??

it = ??

Evaluation of Lappin and Leass 1994

- Weights were arrived at by experimentation on a corpus of computer training manuals.

- Combined with other filters, algorithm achieve 86% accuracy (74% / 89% inter- / intra-sentential):
  - applied to unseen data of same genre

- Hobbs' algorithm applied to same data is 82% accurate (87% / 81% inter / intra).

Hobbs 1978

Tree search Algorithm

- 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

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.

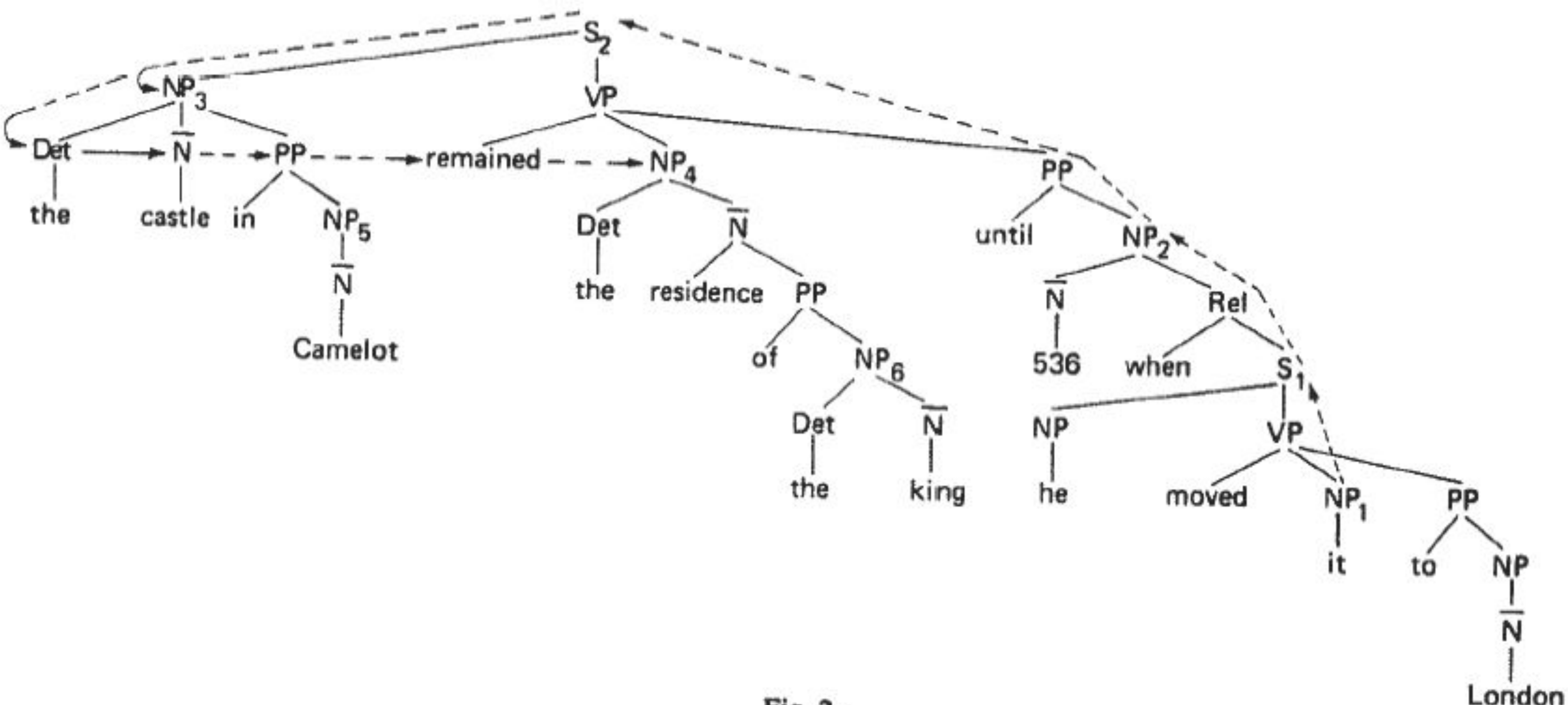


Fig. 2.

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 , e.g. conjoined NPs.

Example:

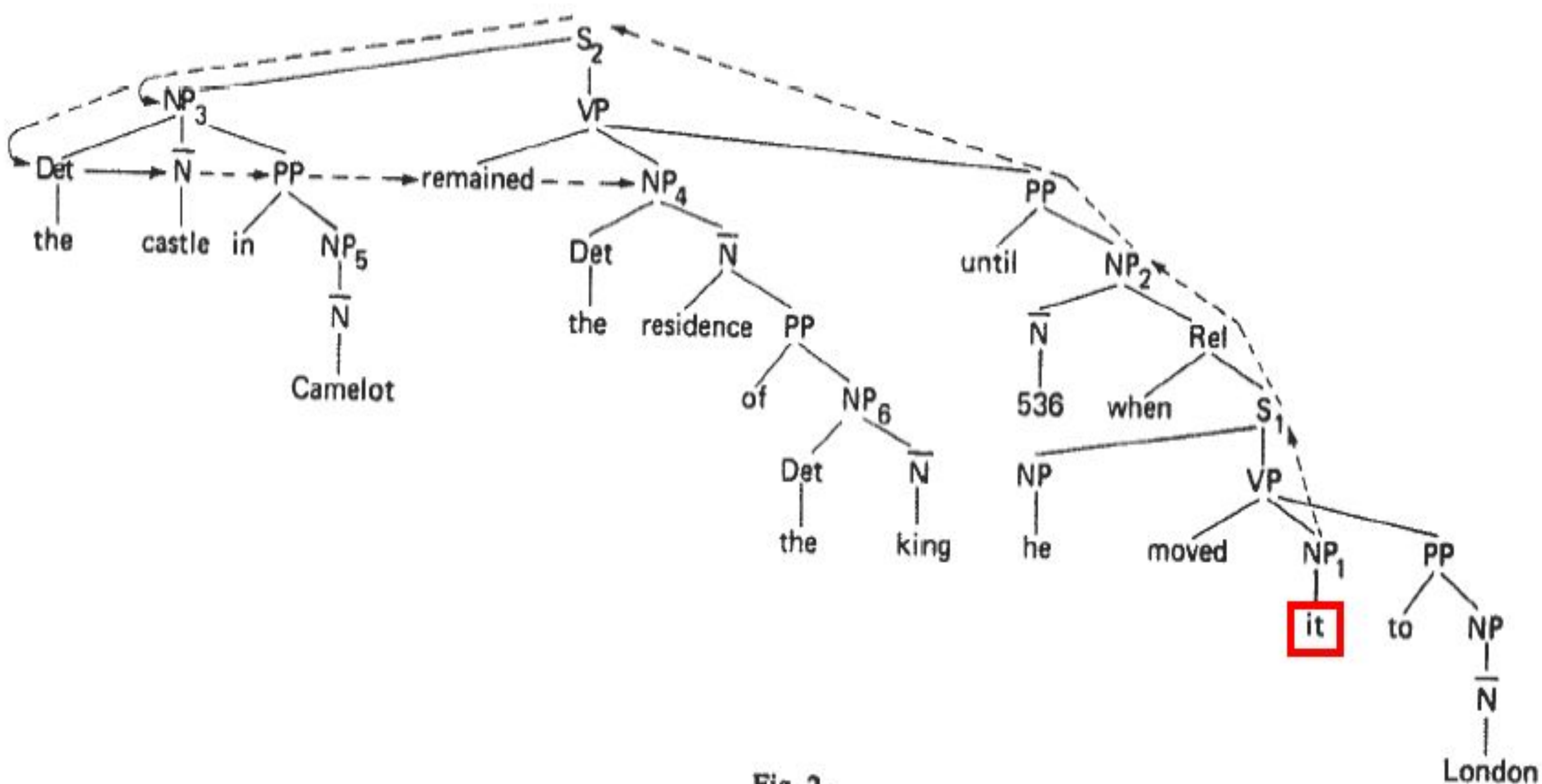


Fig. 2.

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:

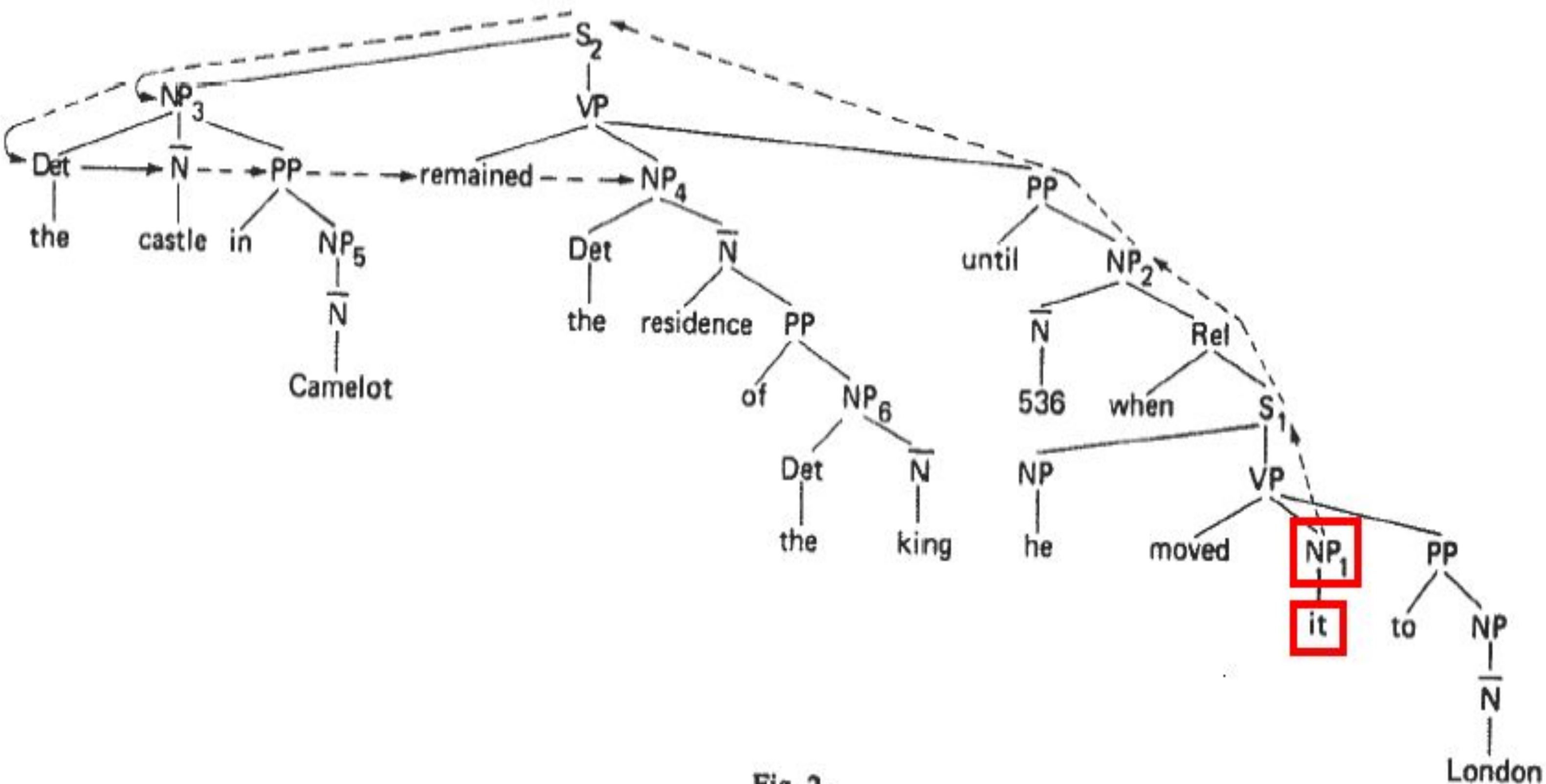


Fig. 2.

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:

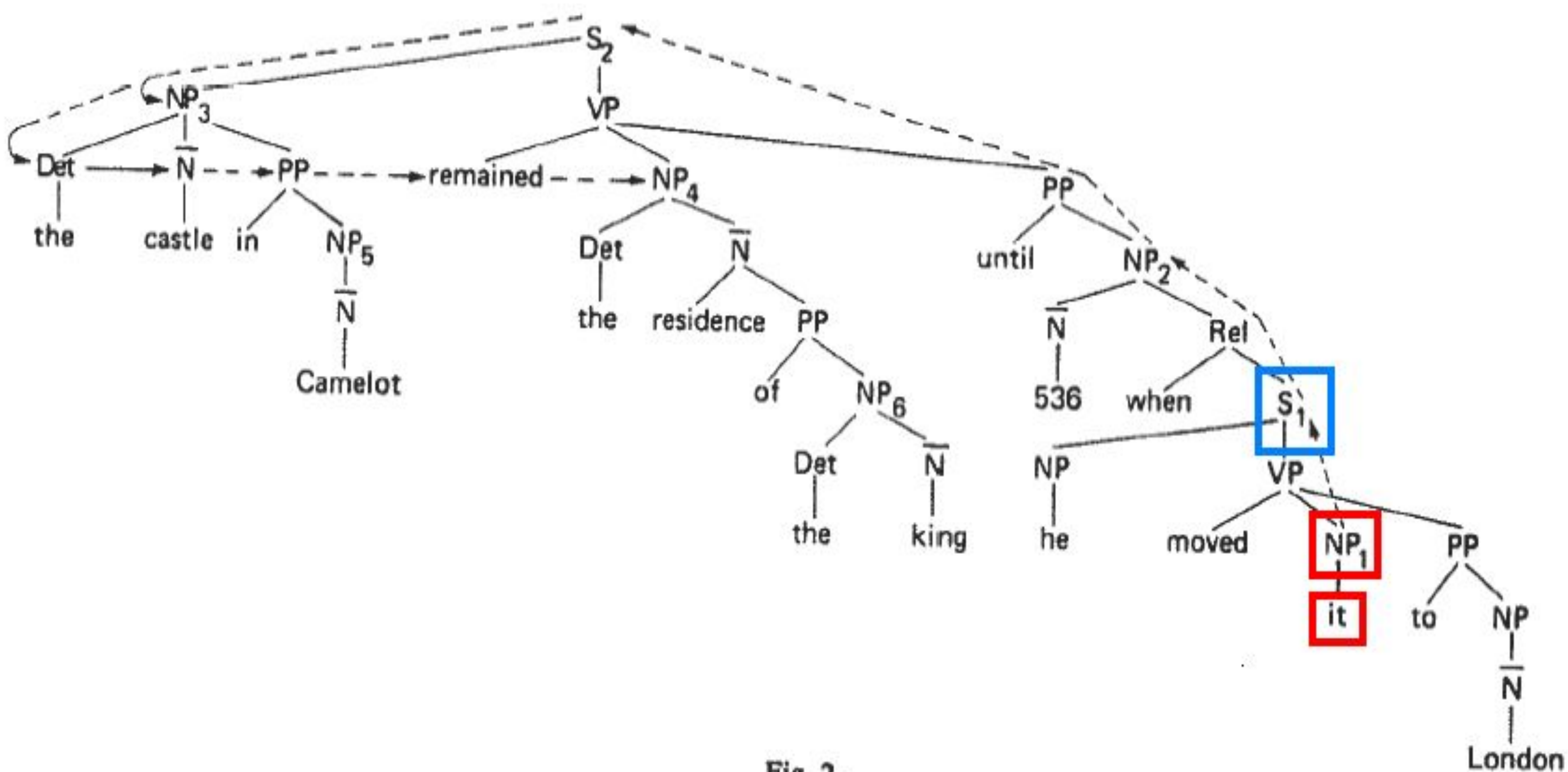


Fig. 2.

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:

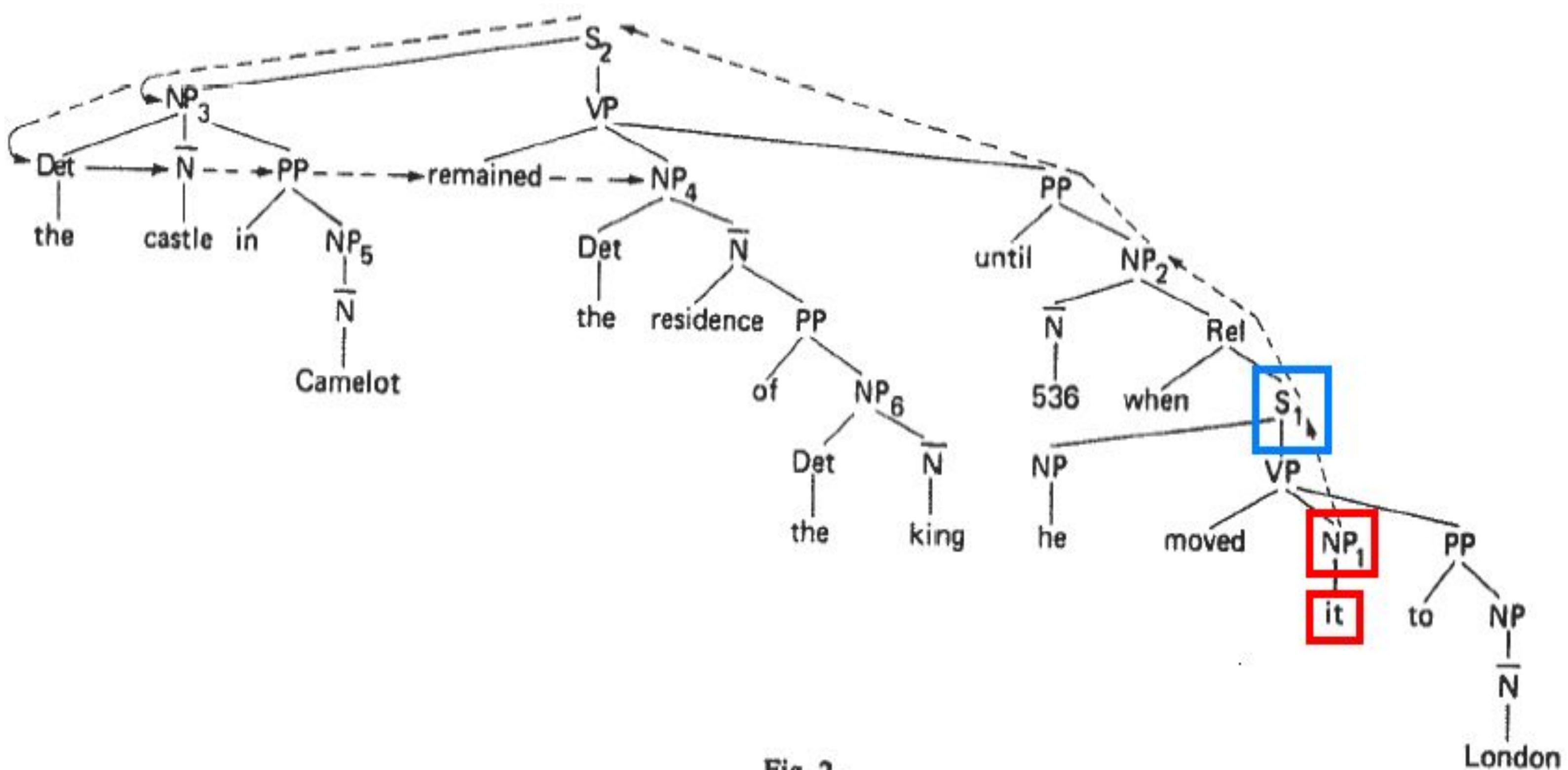


Fig. 2.

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:

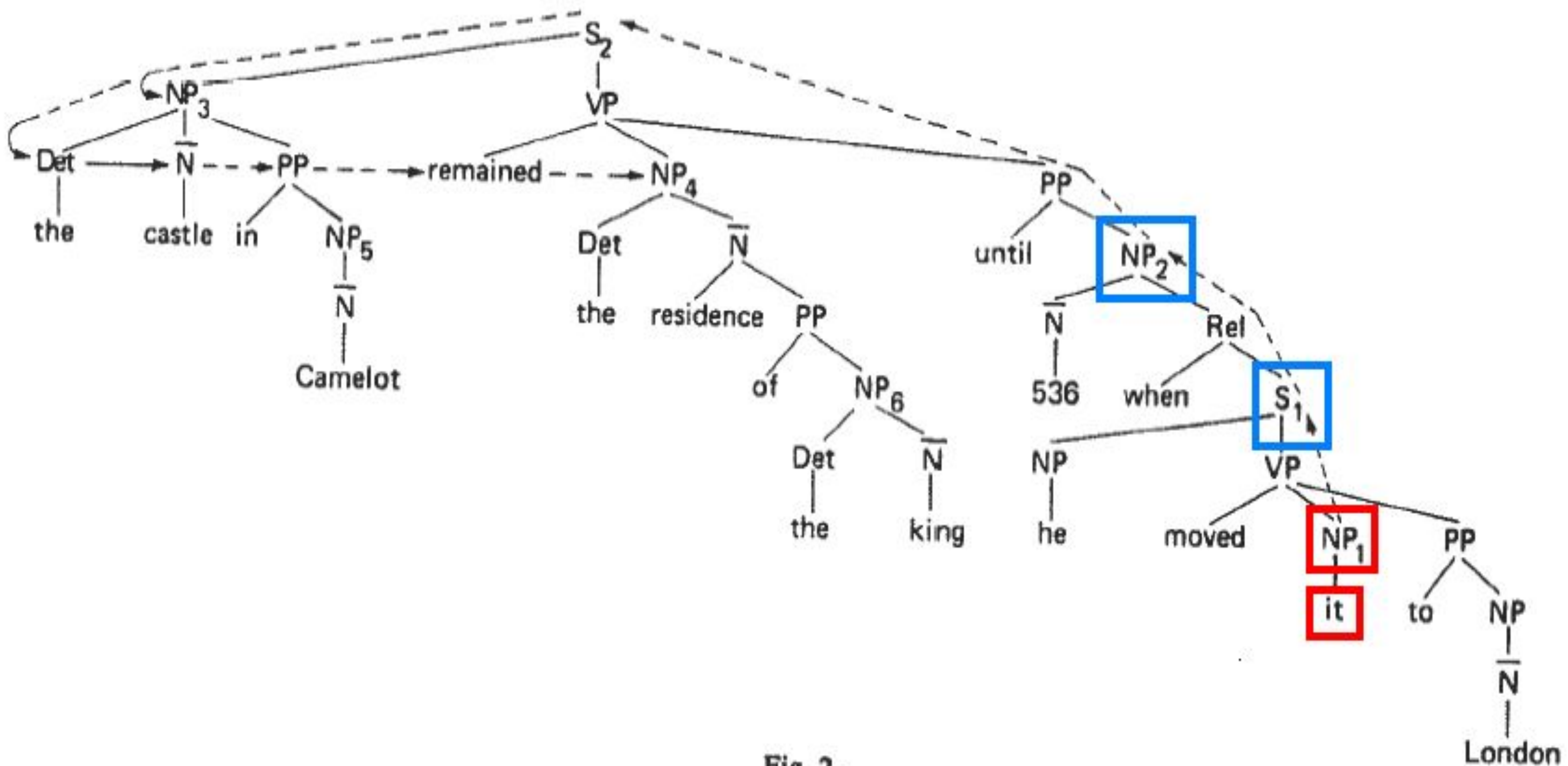


Fig. 2.

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:

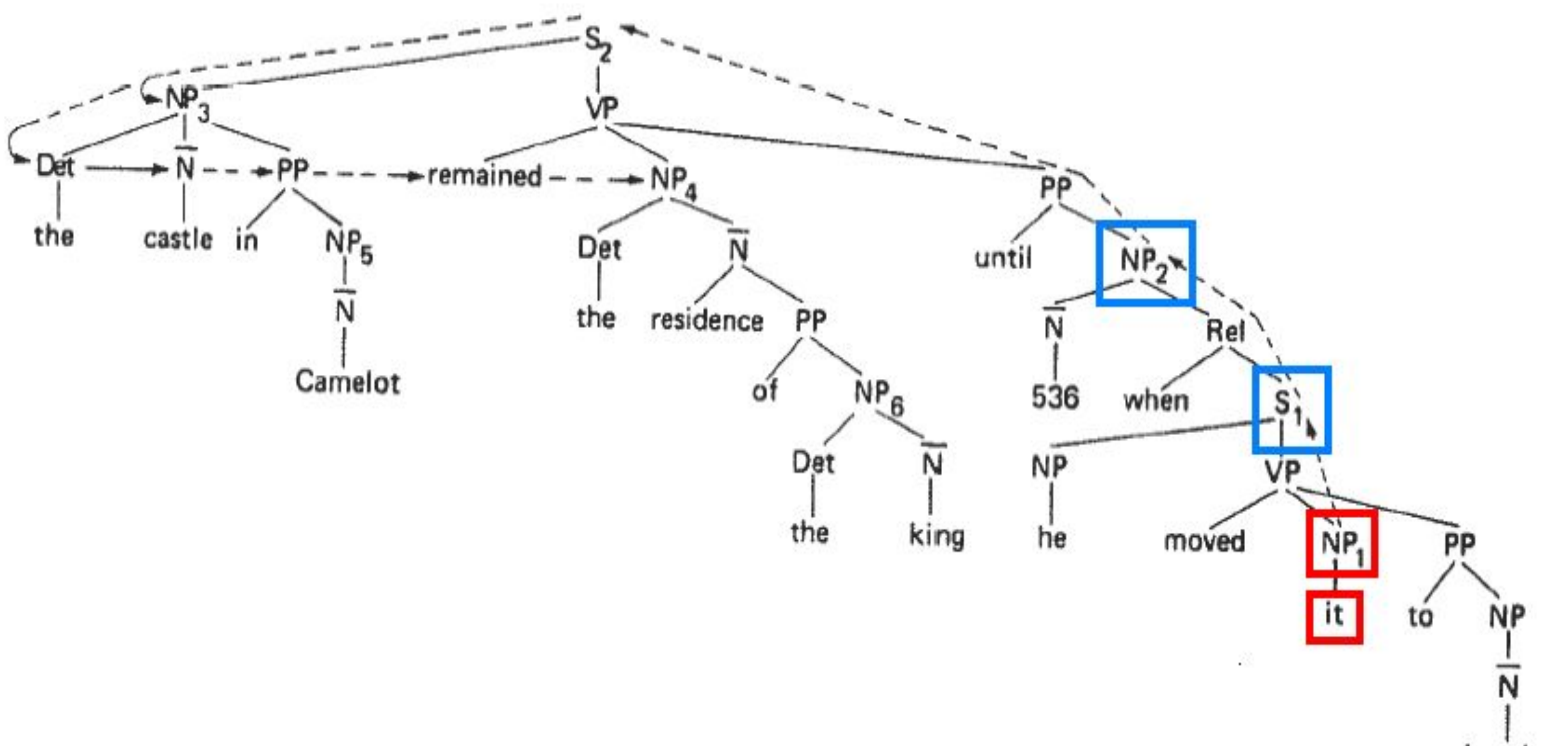


Fig. 2.

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.

Example:

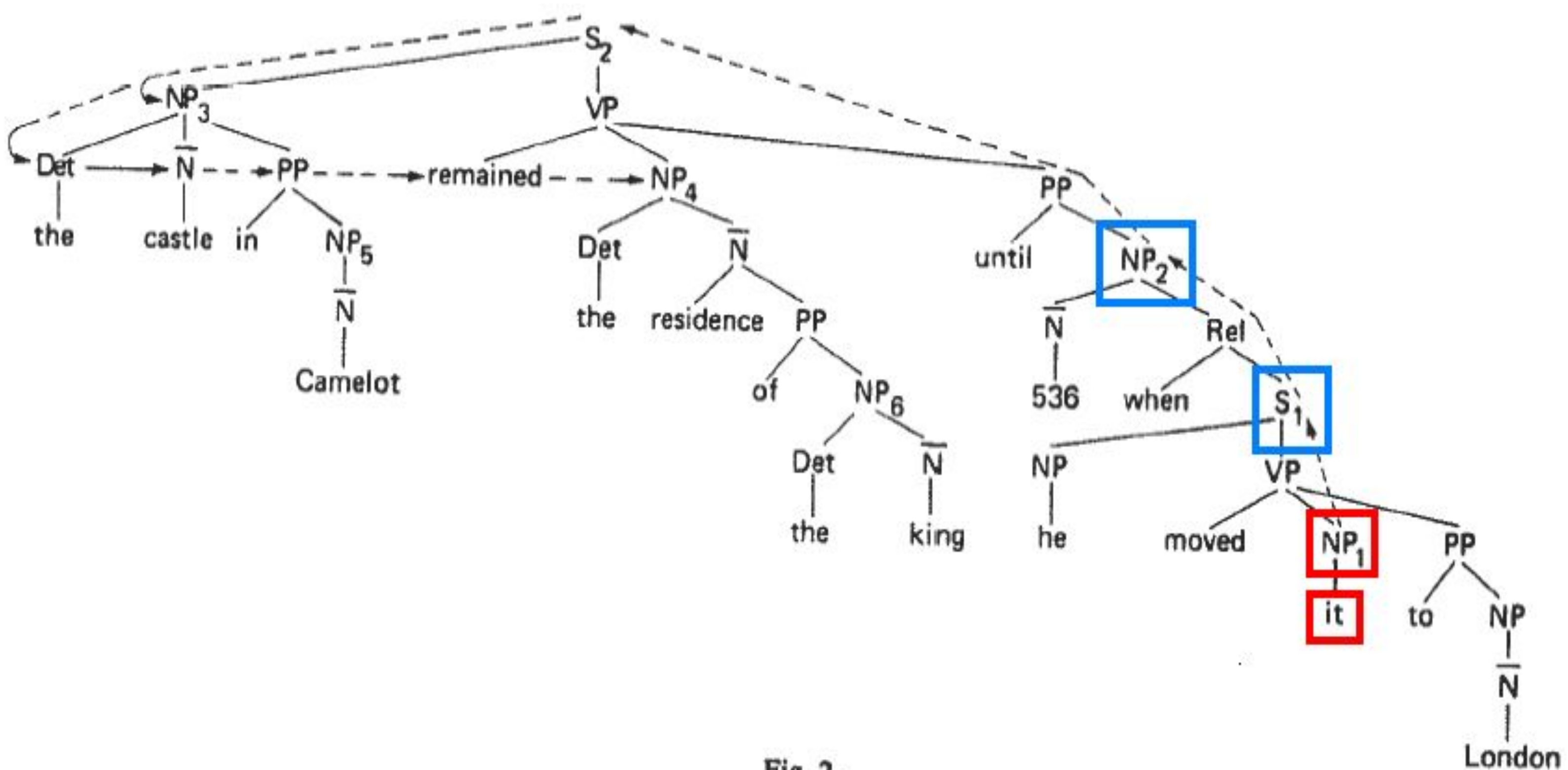


Fig. 2.

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:

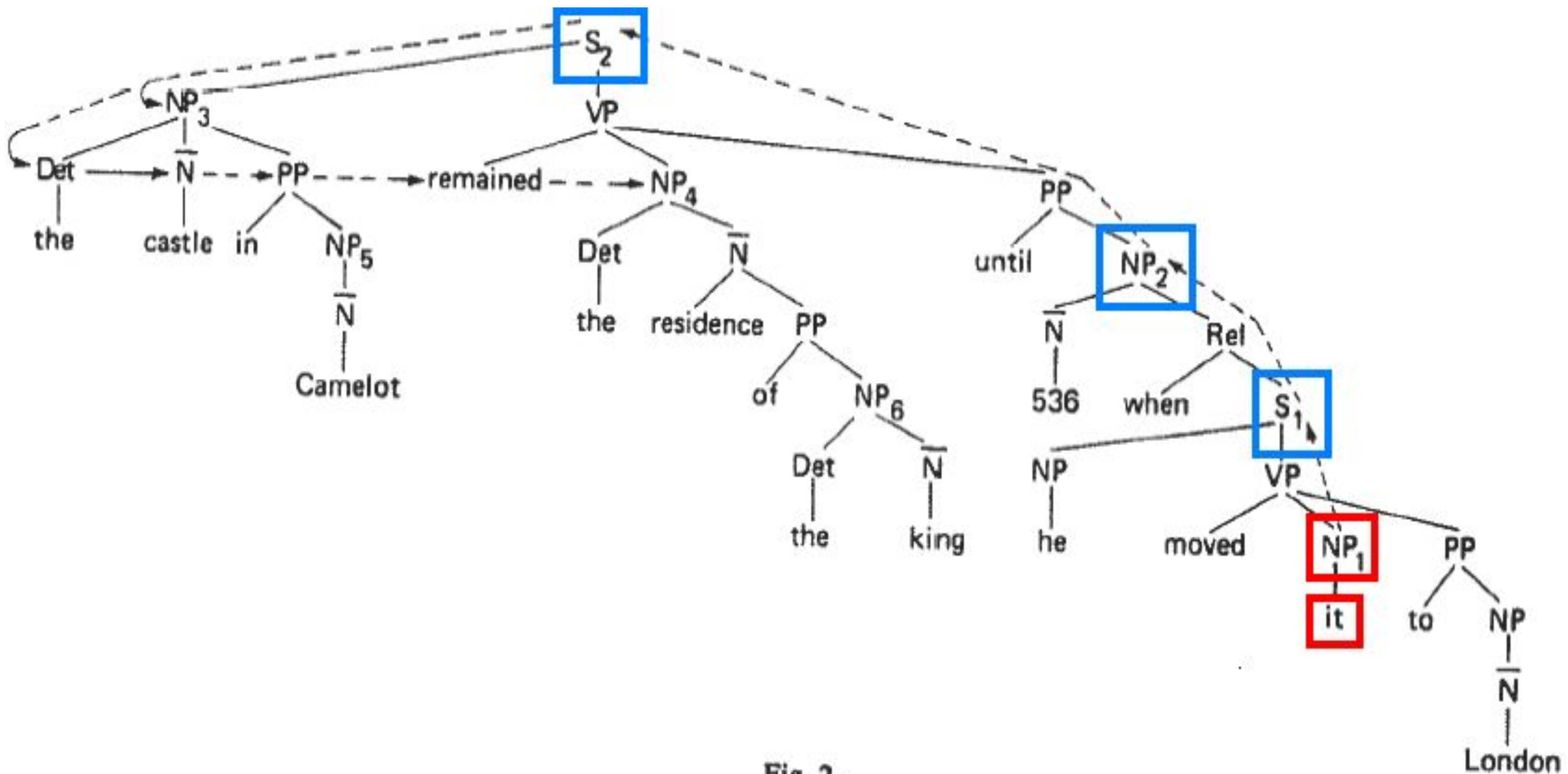


Fig. 2.

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:

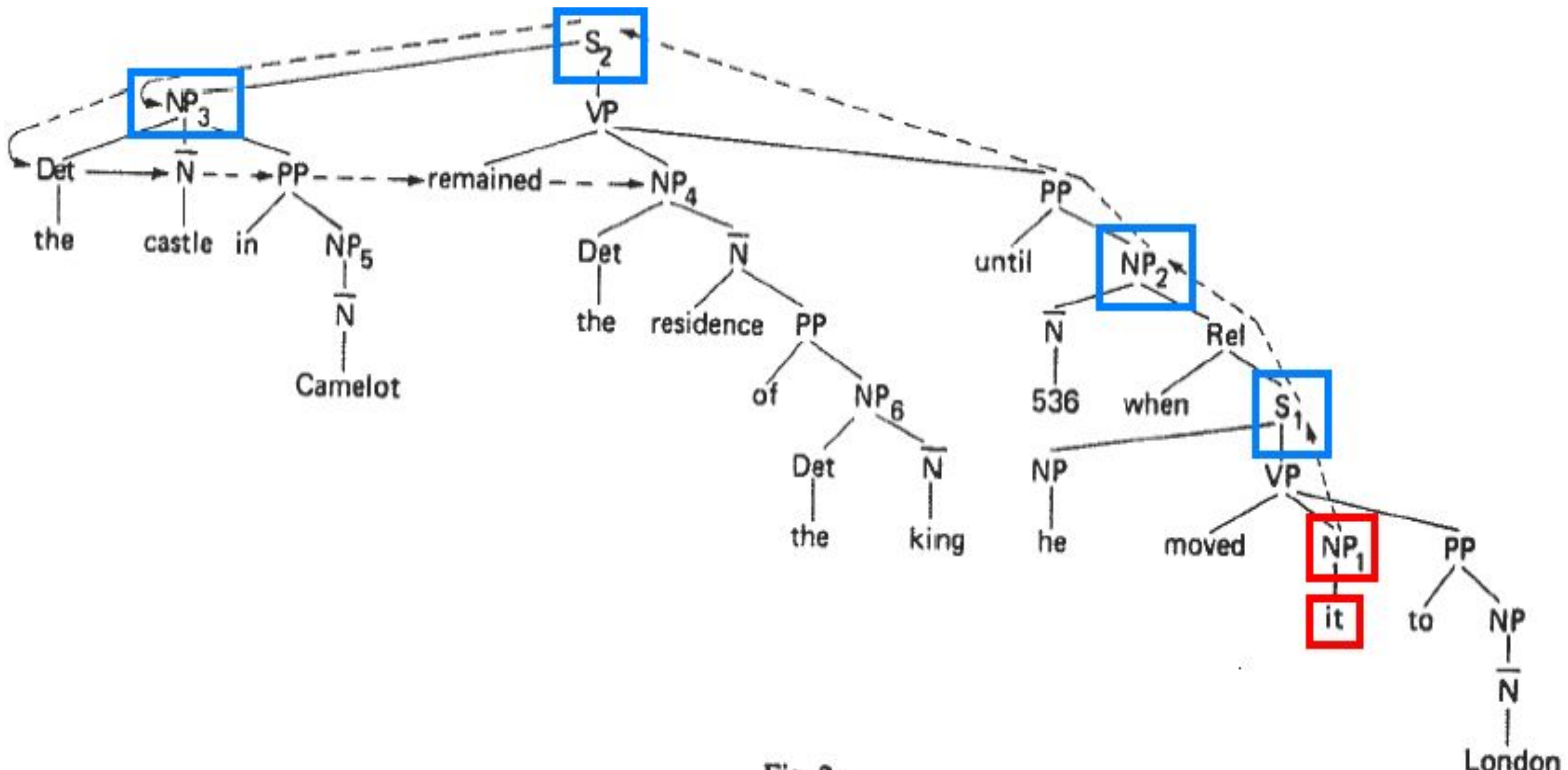


Fig. 2.

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

Example:

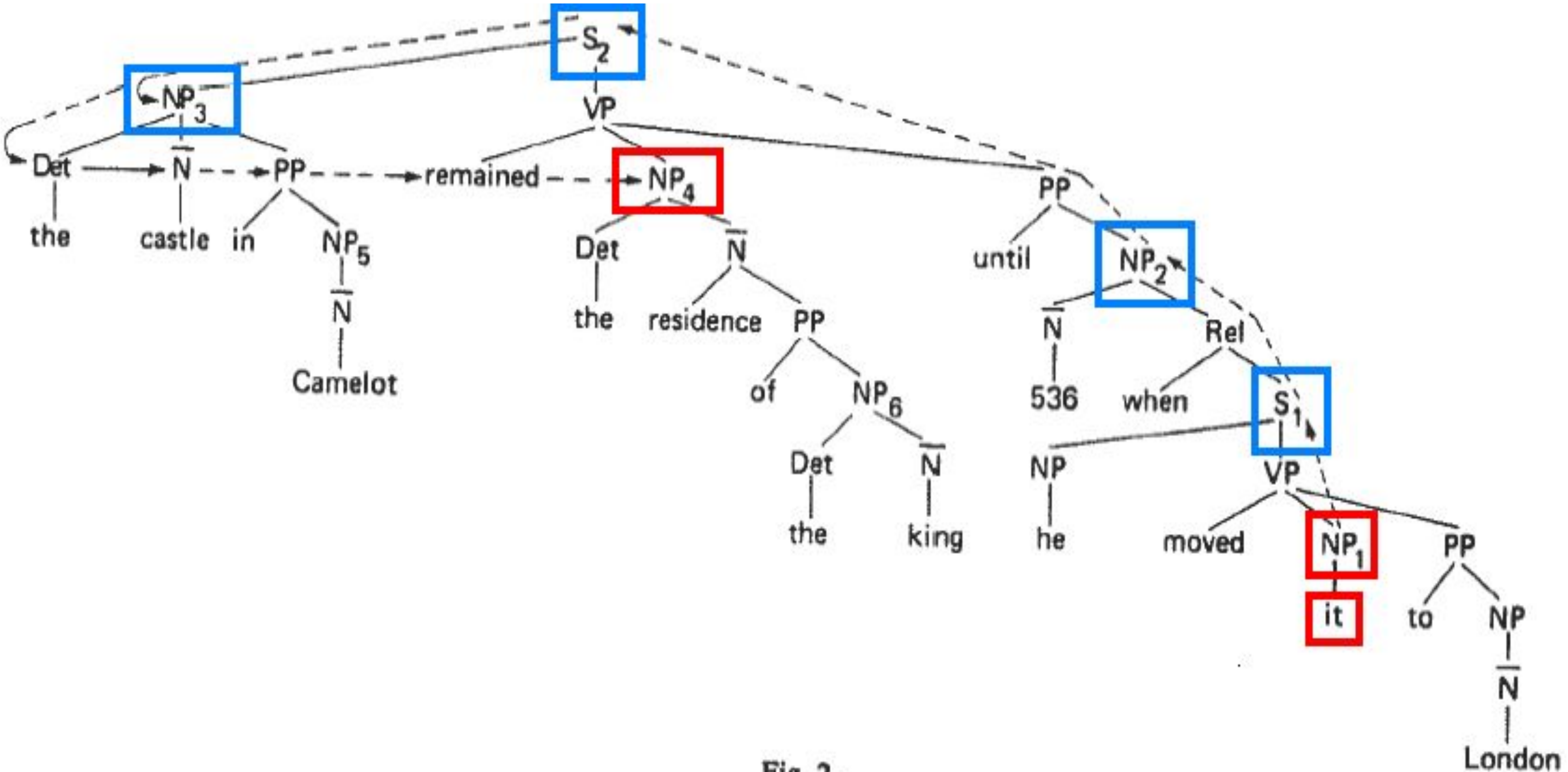


Fig. 2.

NP4: proposed. Accepted.

Another example:

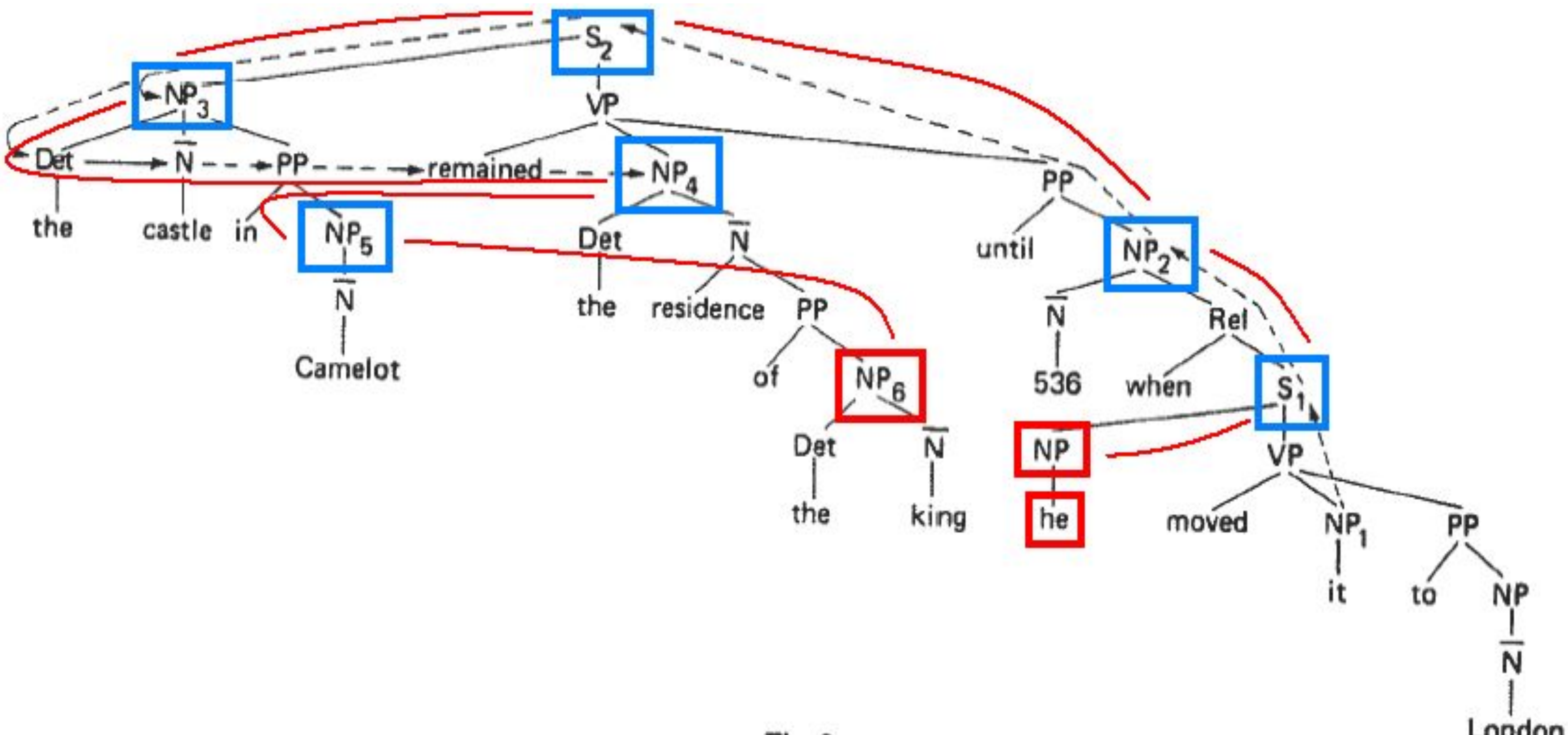


Fig. 2.

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: results

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