Introduction

- Language consists of collocated, related groups of sentences. We refer to such a group of sentences as a **discourse**.

- There are two basic forms of discourse:
  - Monologue;
  - Dialogue;

- We will focus on techniques commonly applied to the interpretation of **monologues**.

Reference Resolution

- **Reference**: the process by which speakers use expressions to denote an entity.
- **Referring expression**: expression used to perform reference .
- **Referent**: the entity that is referred to.
- **Coreference**: referring expressions that are used to refer to the same entity.
- **Anaphora**: reference to a previously introduced entity.

Reference Resolution

- **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.

Reference Phenomena

Five common types of referring expression	
Type	**Example**
Indefinite noun phrase	I saw **a** Ford Escort today.
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 Escort today. Now I want **one**.
Three types of referring expression that complicate the reference resolution	
Type	**Example**
Inferrables	I almost bought a Ford Escort, but a **door** had a dent.
Discontinuous Sets	John and Mary love their Escorts. **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. They are red.
- **Gender Agreement**:
  - To distinguish male, female, and non-personal genders.
    - John has a new car. It is attractive. [It = the new car]
- **Person and Case Agreement**:
  - To distinguish between three forms of person;
    - *You and I have Escorts. They 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.

Syntax can’t be all there is

- John hit Bill. He was severely injured.
- Margaret Thatcher admires Hillary Clinton, and George W. Bush absolutely worships her.

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]

Preferences in Pronoun Interpretation

- **Parallelism (more generally – discourse structure)**:
  - There are also strong preferences that appear to be induced by parallelism effects.
    - Mary went with Sue to the cinema. Sally went with her to the mall. [her = Sue]
    - Jim surprised Paul and then Julie shocked him. (*him* = Paul)

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.

Algorithms

- Hobbs
- Lappin and Leass

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 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.

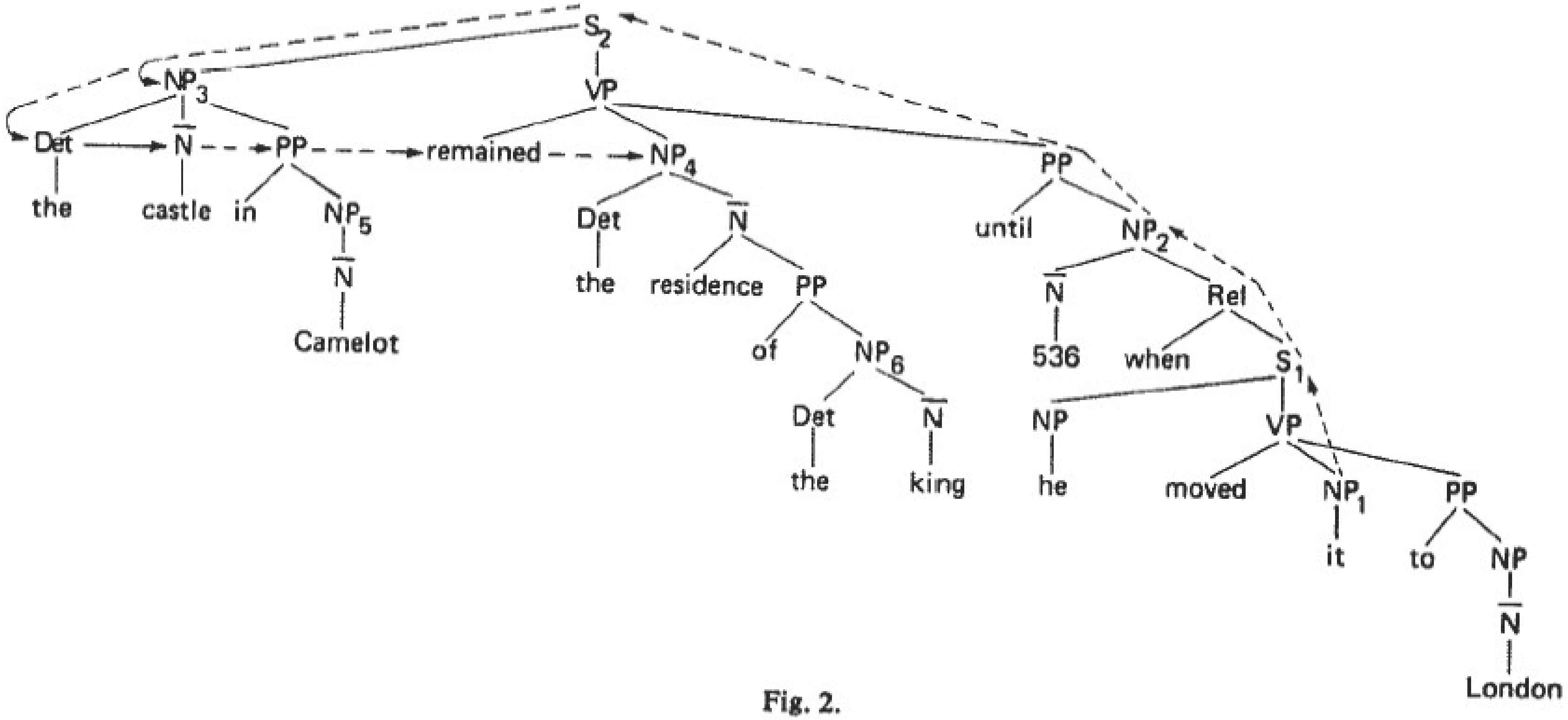


Fig. 2.

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:

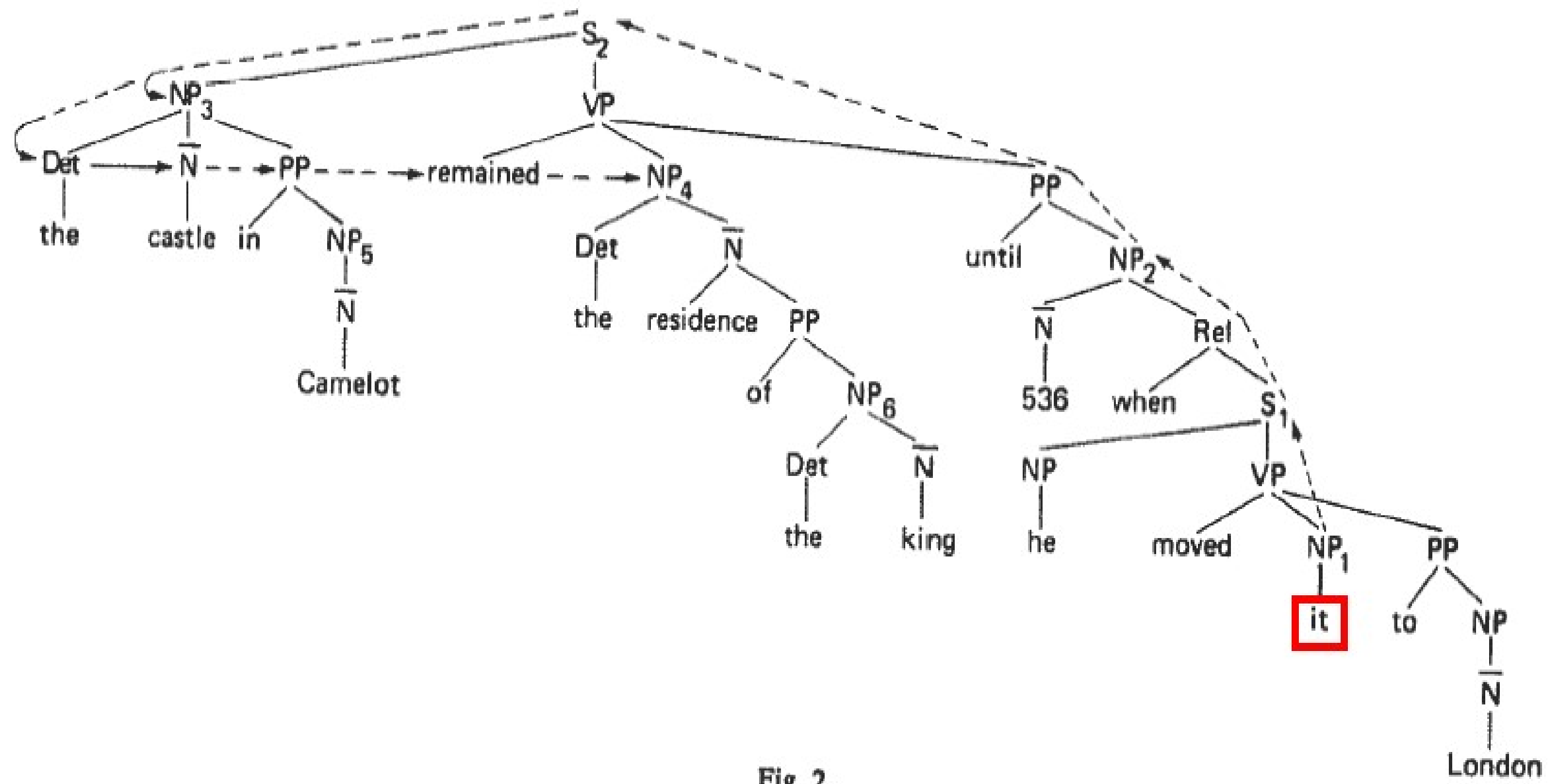


Fig. 2.

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:

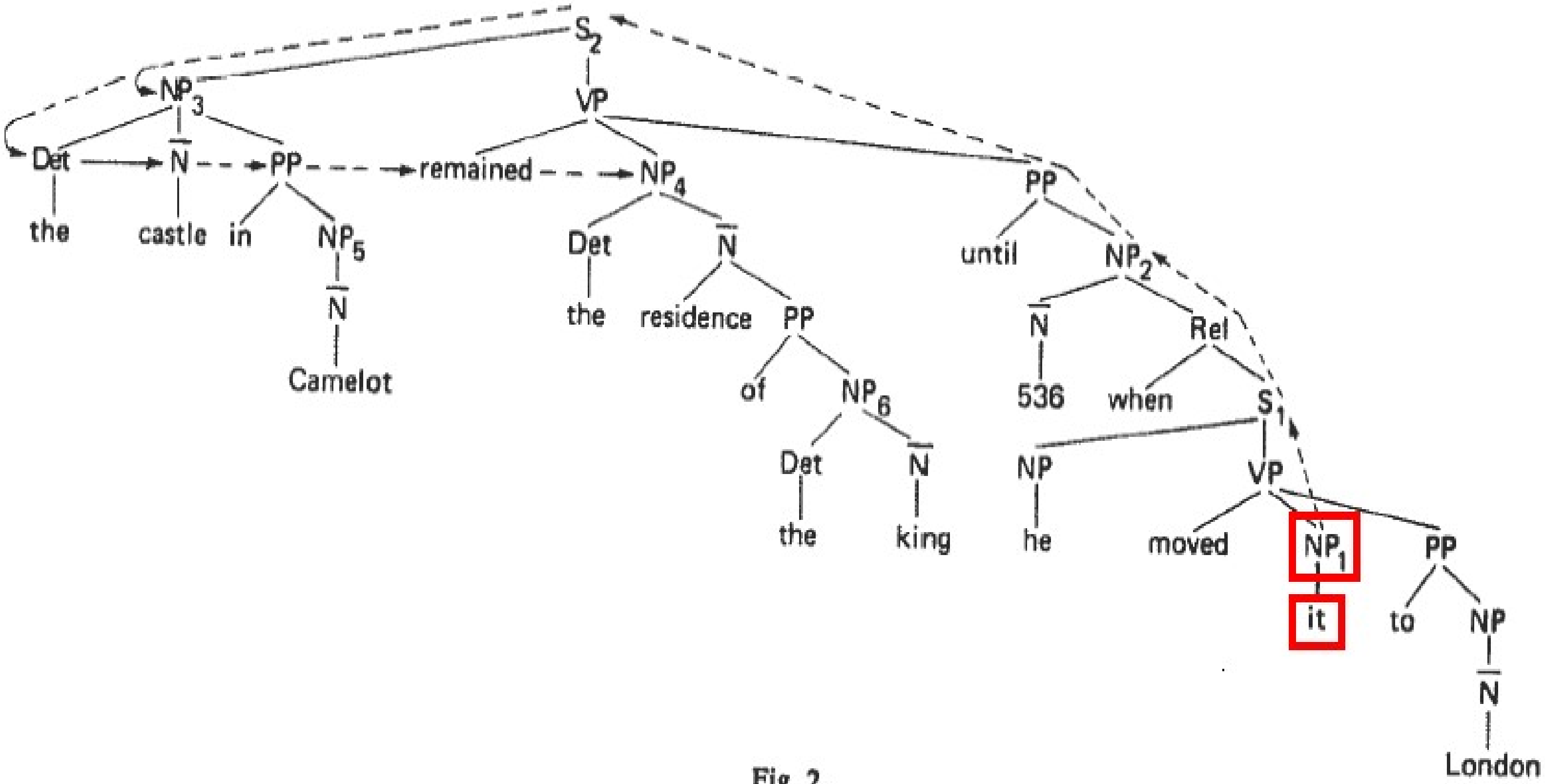


Fig. 2.

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:

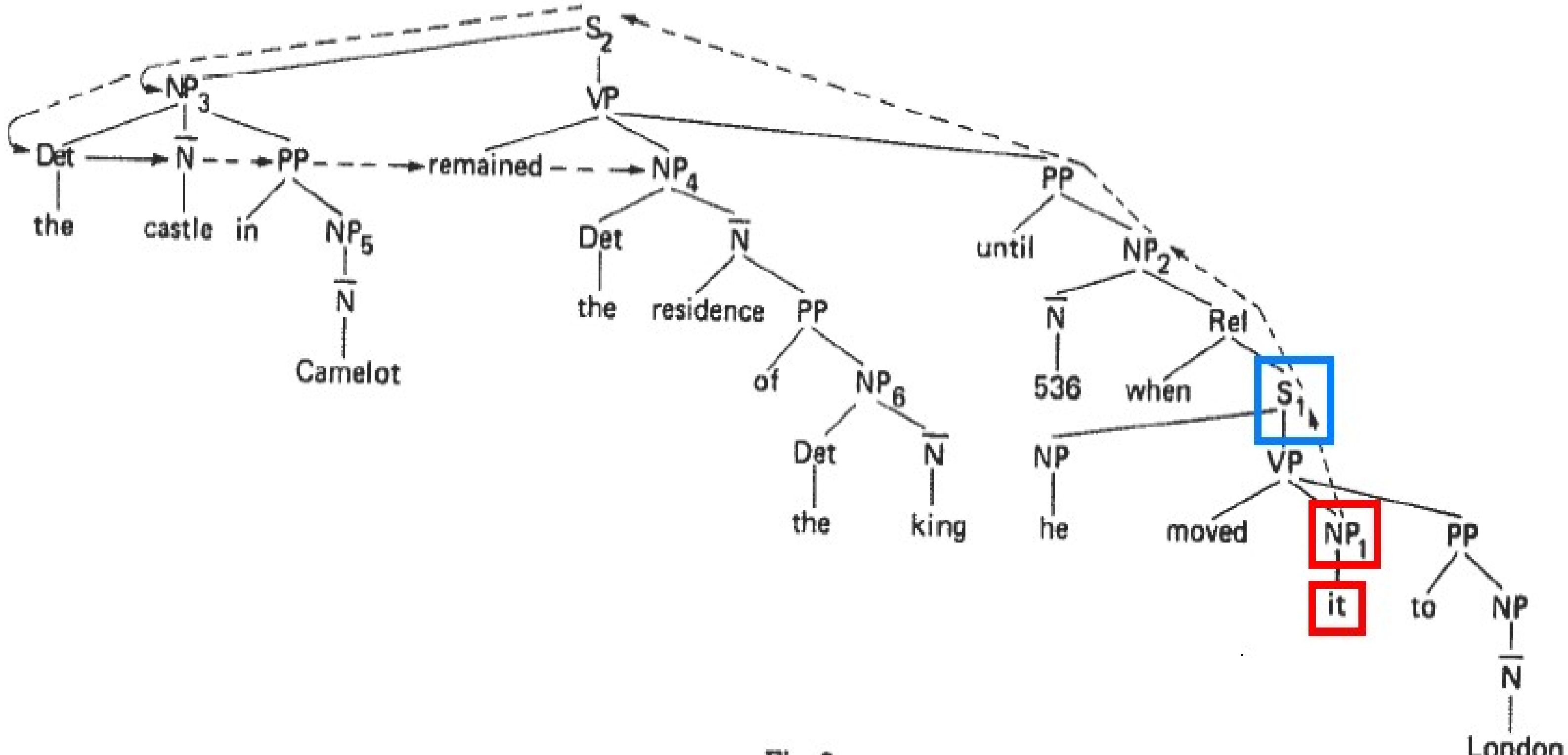


Fig. 2.

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:

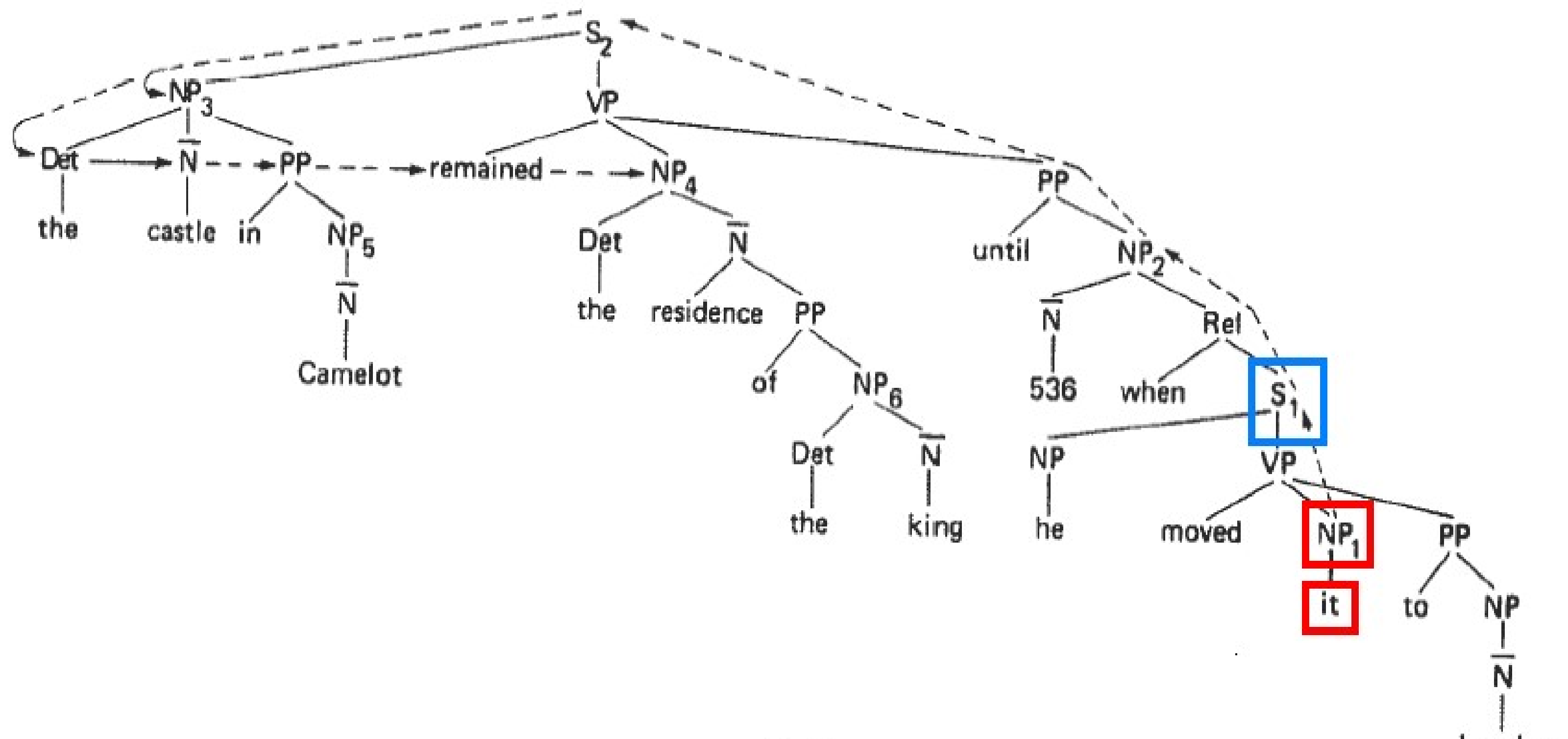


Fig. 2.

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:

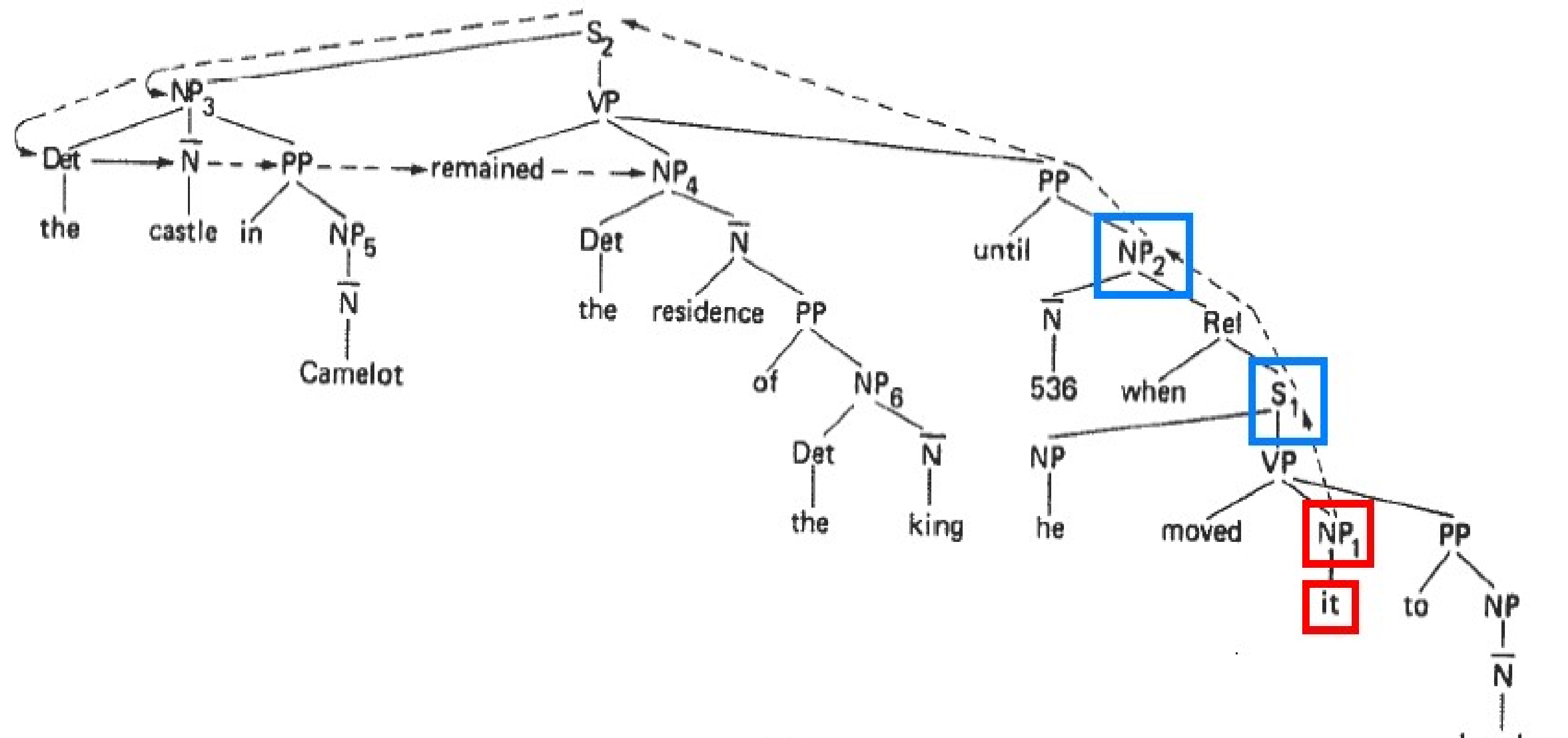


Fig. 2.

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:

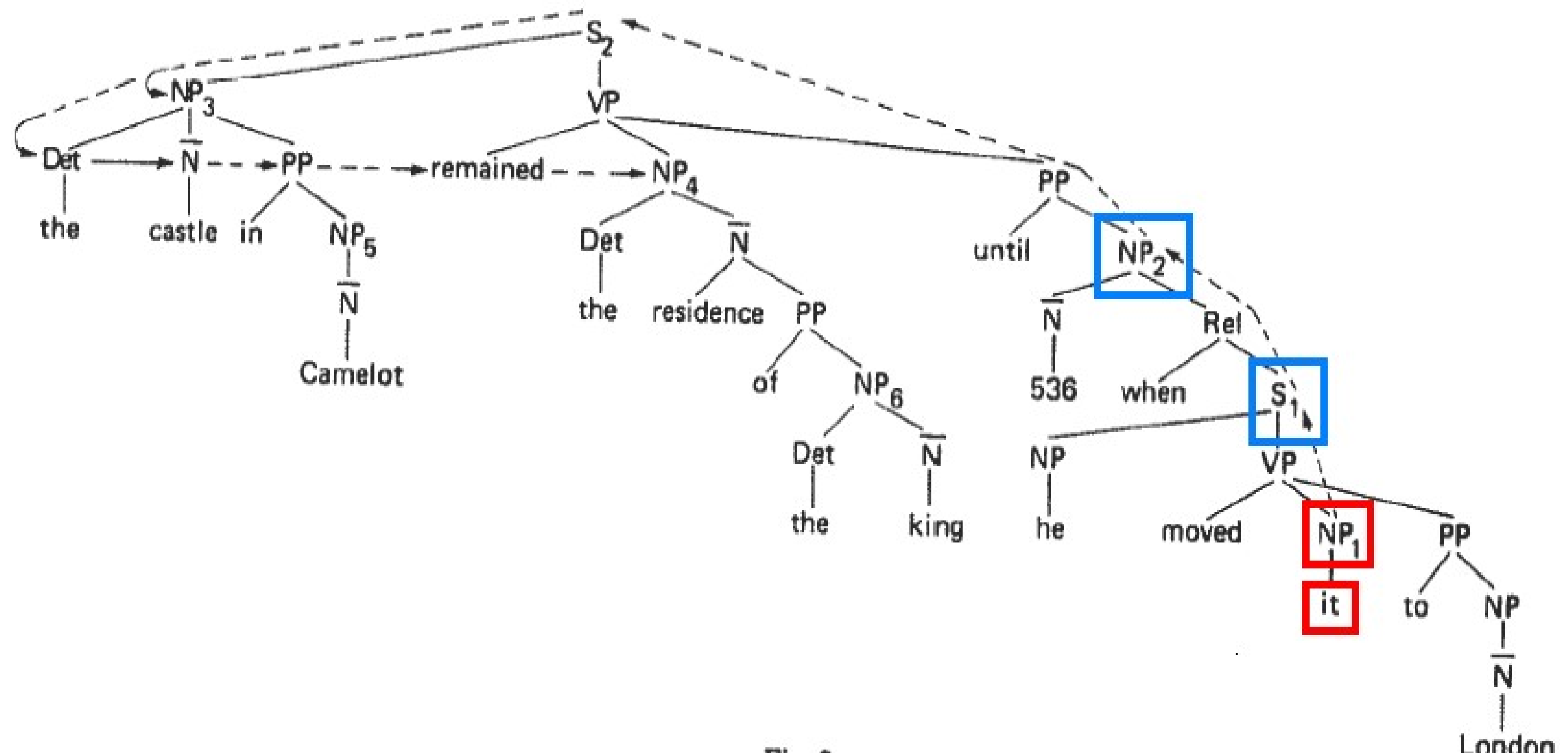


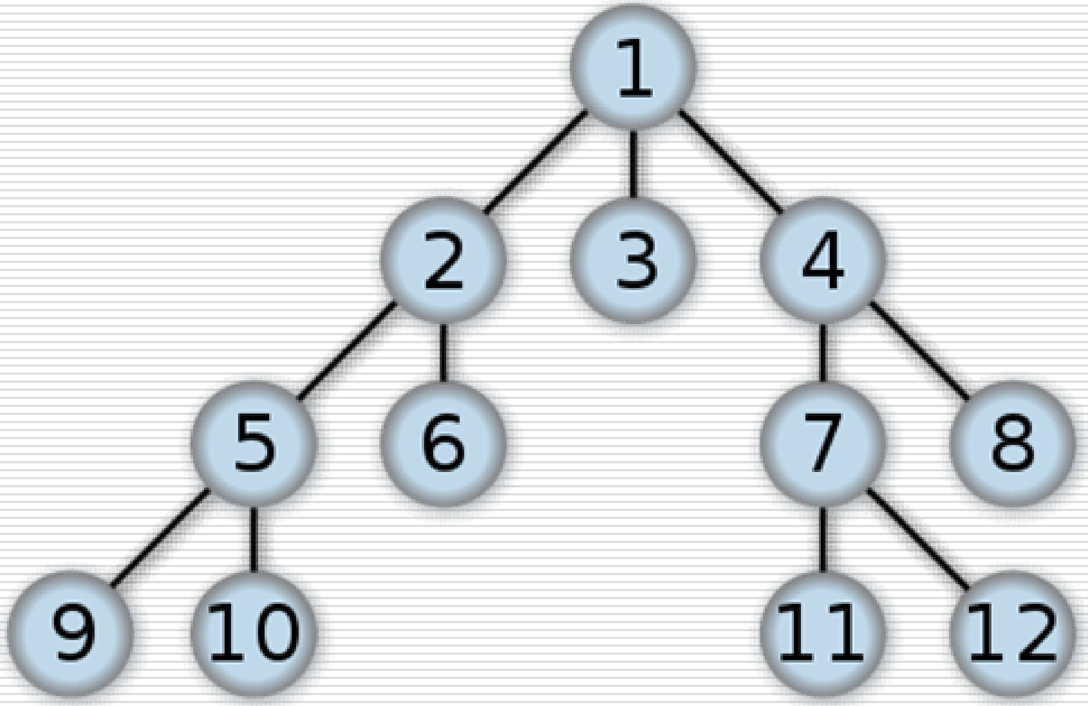
Fig. 2.

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:

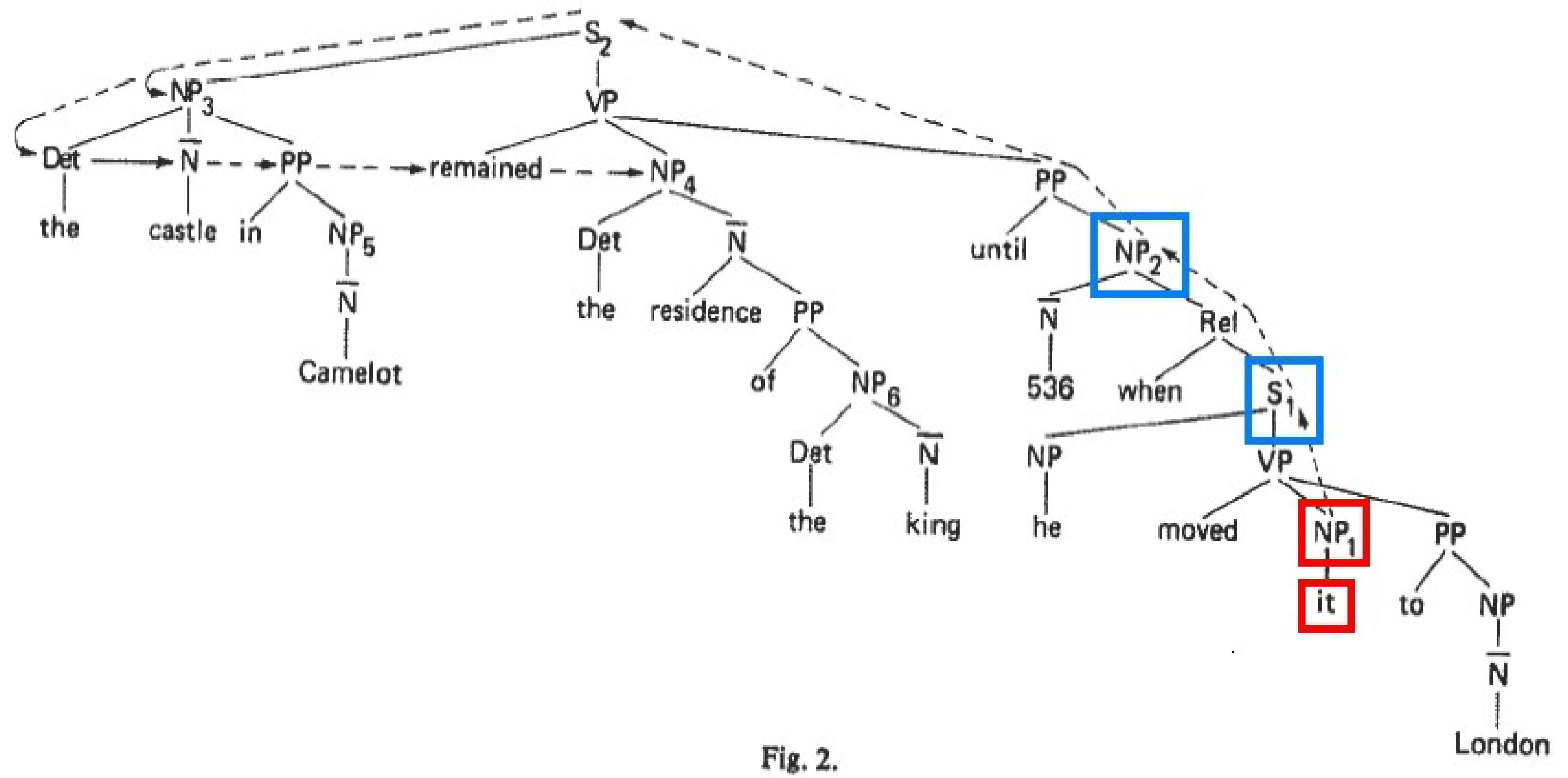


Fig. 2.

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:

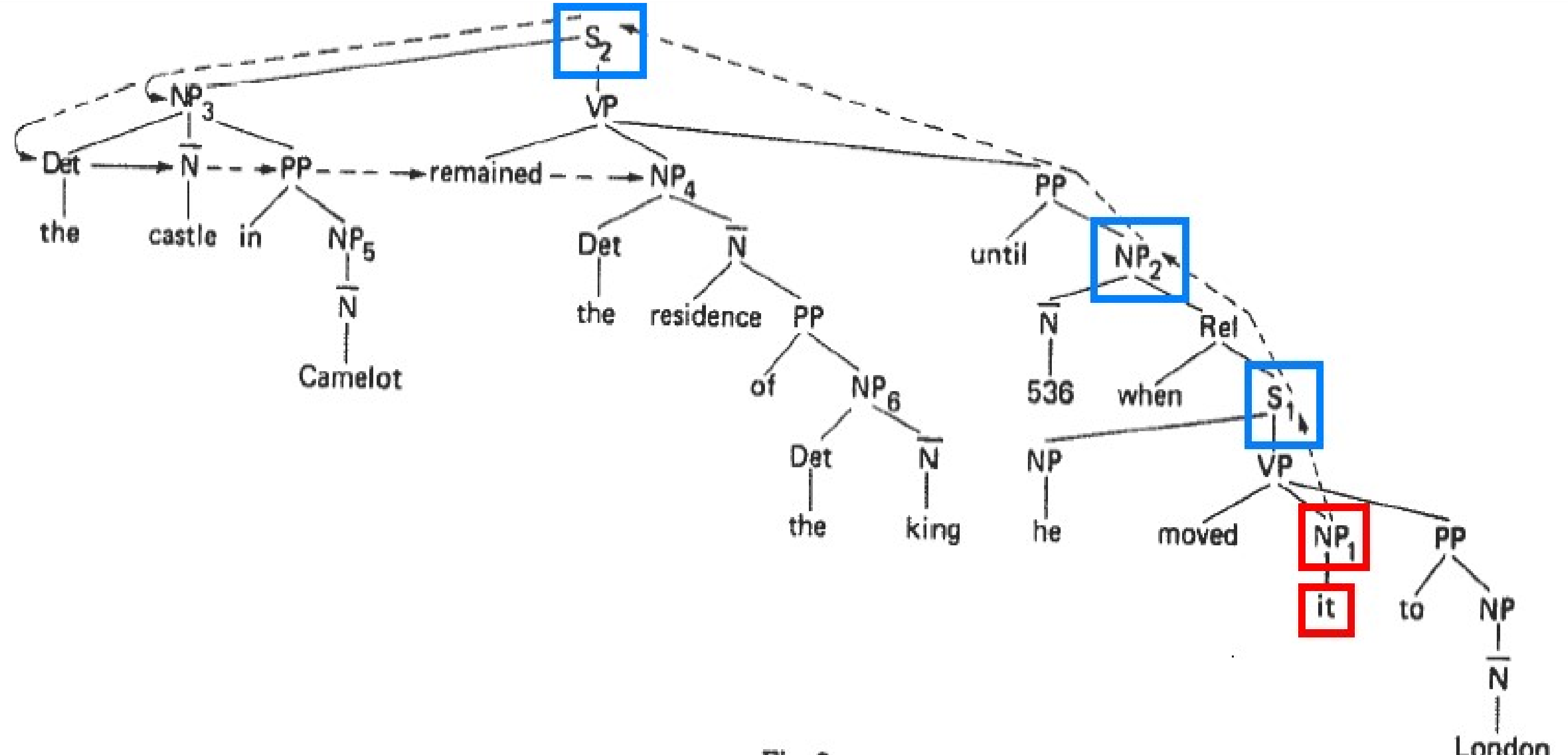


Fig. 2.

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:

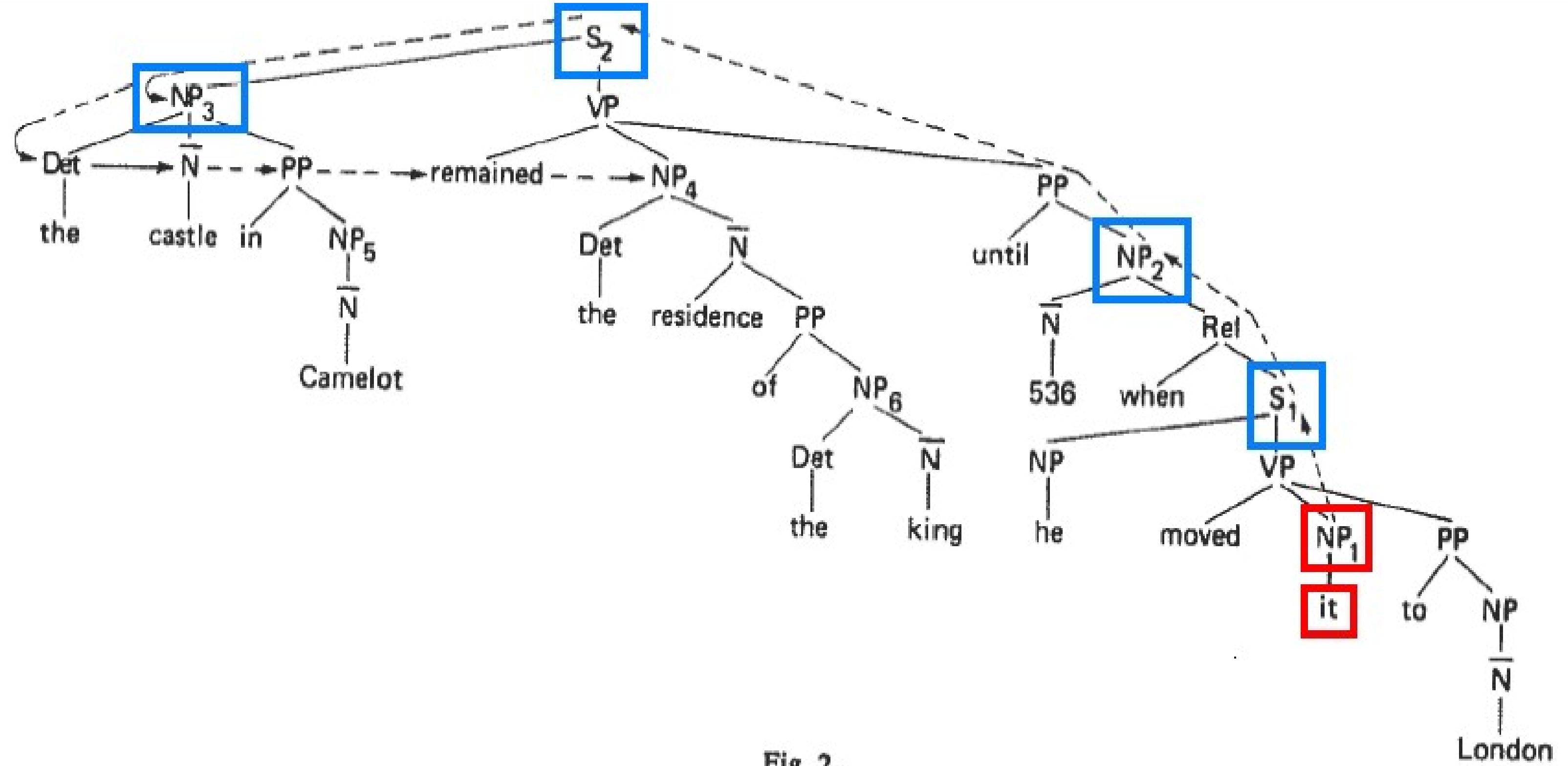


Fig. 2.

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

Example:

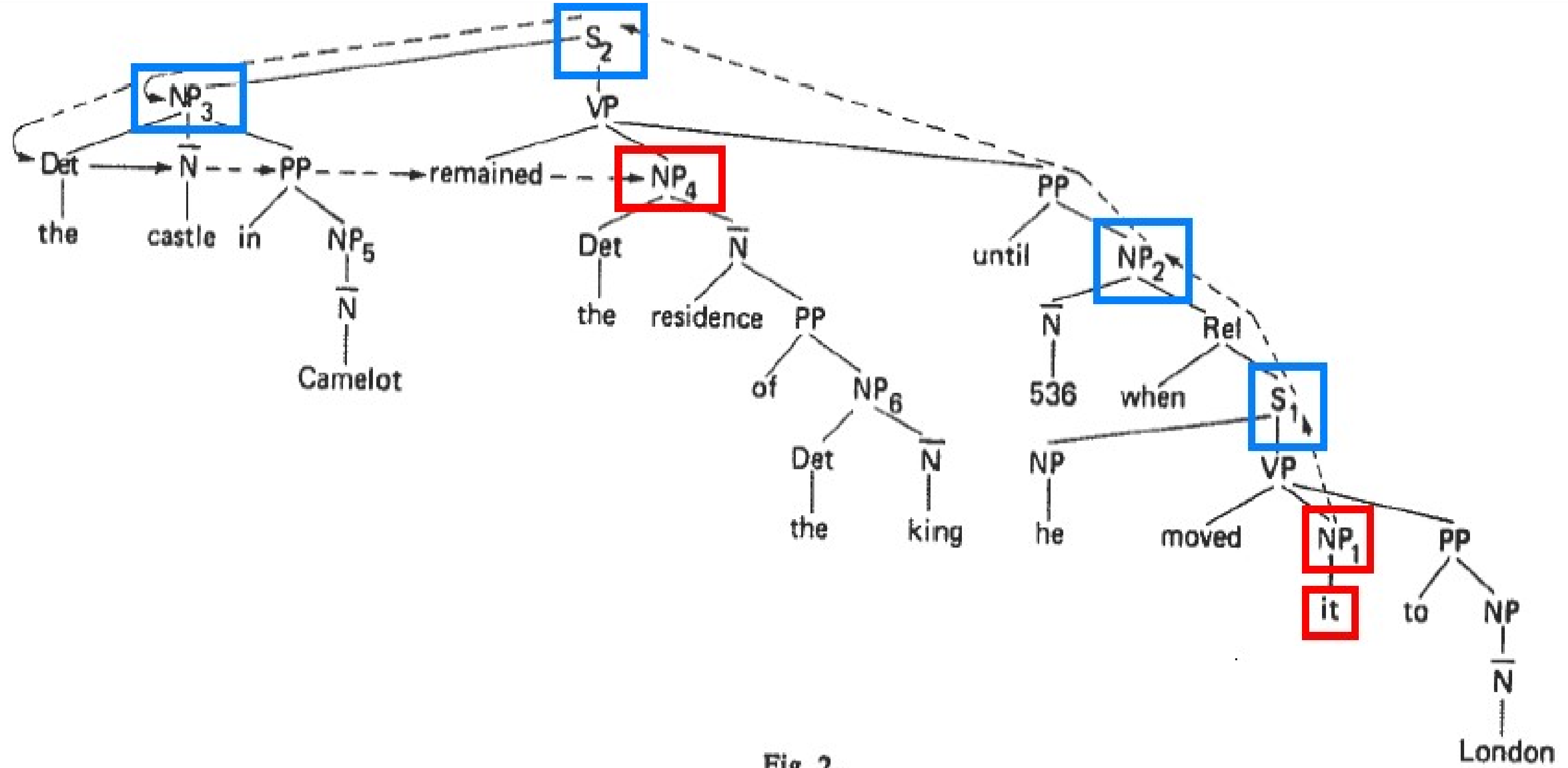


Fig. 2.

NP4: proposed. Accepted.

Another example:

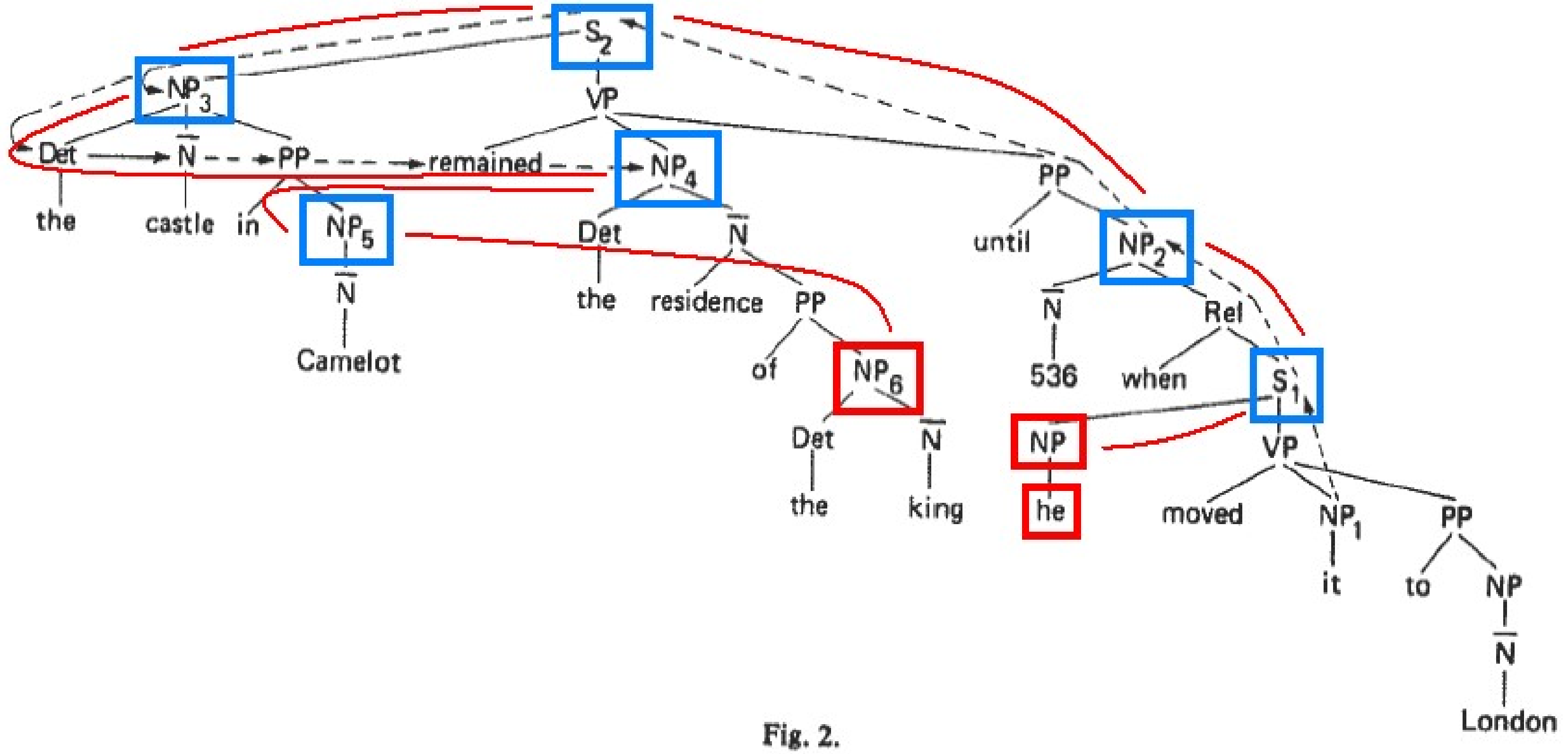


Fig. 2.

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

- 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.

Lappin and Leass 1994

- Idea: Maintain a discourse model , in which there are representations for potential referents. (much like the DRSs we built throughout the quarter)
- 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

Lappin and Leass 1994

- Non-adverbial emphasis is to penalize "demarcated adverbial PPs" (e.g., "In his hand, …") by giving points to all other types.
- Head noun emphasis is to penalize embedded referents.
- Other factors & values:
  - Grammatical role parallelism: 35
  - Cataphora: -175

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.

Lappin and Leass 1994

- Salience factors apply per NP, i.e., referring expression.
- However, we want the salience for a potential referent.
  - So, all NPs determined to have the same referent are examined.
- The referent is given the sum of the highest salience factor associated with any such referring expression.
- Salience factors are considered to have scope over a sentence
  - so references to the same entity over multiple sentences add up
  - while multiple references within the same sentence don't.

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}	?

John

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

Example

- John saw a beautiful Acura Integra at the dealership.

Referent	Phrases	Value
John	{John}	310
Integra	{a beautiful Acura Integra}	?
dealership	{the dealership}	?

Integra

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

Example

- John saw a beautiful Acura Integra at the dealership.

Referent	Phrases	Value
John	{John}	310
Integra	{a beautiful Acura Integra}	280
dealership	{the dealership}	?

dealership

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

Example

- John saw a beautiful Acura Integra at the dealership.

Referent	Phrases	Value
John	{John}	310
Integra	{a beautiful Acura Integra}	280
dealership	{the dealership}	230

Example

- He showed it to Bob.

Referent	Phrases	Value
John	{John}	310/2
Integra	{a beautiful Acura Integra}	280/2
dealership	{the dealership}	230/2

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

He

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

Example

- He showed it to Bob.

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

It

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

Example

- He showed it to Bob.

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.

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

Bob

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

Example

- He showed it to Bob.

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}	465/2
Integra	{a beautiful Acura Integra, it_1}	420/2
Bob	{Bob}	270/2
dealership	{the dealership}	115/2

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

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

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

Since “John” is more salient than “Bob” (232.5>135):

- “he” refers to “John”

Example

- He bought it.

Referent	**Phrases**	**Value**
John	{John, he_1,he_2}	542.5
Integra	{a beautiful Acura Integra, it_1}	210
Bob	{Bob}	135
dealership	{the dealership}	57.5

It

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

Example

- He bought it.

Referent	**Phrases**	**Value**
John	{John, he_1,he_2}	542.5
Integra	{a beautiful Acura Integra, it_1}	210
Bob	{Bob}	135
dealership	{the dealership}	57.5

Since "Integra" is more salient than "dealership" (210>57.5):

- "it" refers to "Integra"

Example

- He bought it.

Referent	Phrases	Value
John	{John, he_1,he_2}	542.5
Integra	{a beautiful Acura Integra, it_1,it_2}	490
Bob	{Bob}	135
dealership	{the dealership}	57.5

We should have added 35 for grammatical role parallelism, but we ignore this.

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).

Mention Pair Architecture

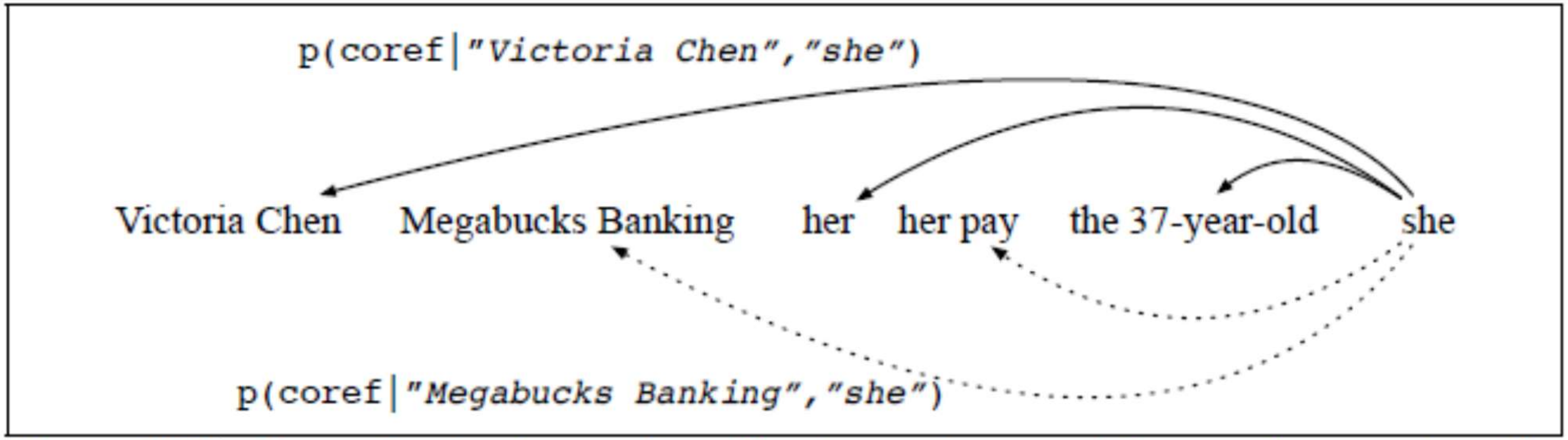


Figure 22.2 For each pair of a mention (like *she*), and a potential antecedent mention (like *Victoria Chen* or *her*), the mention-pair classifier assigns a probability of a coreference link.

Features of the Anaphor or Antecedent Mention		
First (last) word	Victoria/she	First or last word (or embedding) of antecedent/anaphor
Head word	Victoria/she	Head word (or head embedding) of antecedent/anaphor
Attributes	Sg-F-A-3-PER/Sg-F-A-3-PER	The number, gender, animacy, person, named entity type attributes of (antecedent/anaphor)
Length	2/1	length in words of (antecedent/anaphor)
Grammatical role	Sub/Sub	The grammatical role—subject, direct object, indirect object/PP—of (antecedent/anaphor)
Mention type	P/Pr	Type: (P)roper, (D)efinite, (I)ndefinite, (Pr)onoun) of antecedent/anaphor
Features of the Antecedent Entity		
Entity shape	P-Pr-D	The 'shape' or list of types of the mentions in the antecedent entity (cluster), i.e., sequences of (P)roper, (D)efinite, (I)ndefinite, (Pr)onoun.
Entity attributes	Sg-F-A-3-PER	The number, gender, animacy, person, named entity type attributes of the antecedent entity
Antecedent cluster size	3	Number of mentions in the antecedent cluster
Features of the Pair of Mentions		
Longer anaphor	F	True of anaphor is longer than antecedent
Pairs of any features	Victoria/she, 2/1, Sub/Sub, P/Pr, etc .	For each individual feature, pair of type of antecedent+ type of anaphor
Sentence distance	1	The number of sentences between antecedent and anaphor
Mention distance	4	The number of mentions between antecedent and anaphor
i-within-i	F	Anaphor has i-within-i relation with antecedent
Cosine		Cosine between antecedent and anaphor embeddings
Appositive	F	True if the anaphor is in the syntactic apposition relation to the antecedent. This can be useful even if appositives are not mentions (to know to attach the appositive to a preceding head)
Features of the Pair of Entities		
Exact String Match	F	True if the strings of any two mentions from the antecedent and anaphor clusters are identical.
Head Word Match	F	True if any mentions from antecedent cluster has same headword as any mention in anaphor cluster
Word Inclusion	F	Words in antecedent cluster includes all words in anaphor cluster
Features of the Document		
Genre/source	N	The document genre— (D)ialog, (N)ews, etc,

Figure 22.4 Some common features for feature-based coreference algorithms, with values for the anaphor "she" and potential antecedent "Victoria Chen".