Hobbs Algorithm for Anaphora Resolution

Complete study notes | Natural Language Processing

Definition, overview, architecture, step-by-step workflow, algorithm, pseudocode, worked examples, and applications.

1. Definition

Anaphora resolution identifies the earlier expression needed to interpret a referring expression. In “Ravi entered. He smiled”, the pronoun **He** refers to **Ravi**; Ravi is its antecedent.

Hobbs’ algorithm is a deterministic, syntax-based method for finding antecedents of third-person pronouns. It searches constituency parse trees in a specified order and proposes noun phrases, which are checked for compatibility with the pronoun. It is commonly associated with Jerry R. Hobbs’ 1978 paper [1].

2. Overview and characteristics

• **Rule-based:** no model training is required for the search procedure.

• **Tree-based:** uses phrase structure, especially NP and S nodes.

• **Ordered search:** tree depth and left-to-right ordering determine candidate priority.

• **Context-sensitive:** searches the current sentence before earlier sentences.

• **Compatibility filtering:** agreement and suitable semantic constraints reject inappropriate candidates.

• **Interpretable:** a resolution can be explained by tracing visited nodes.

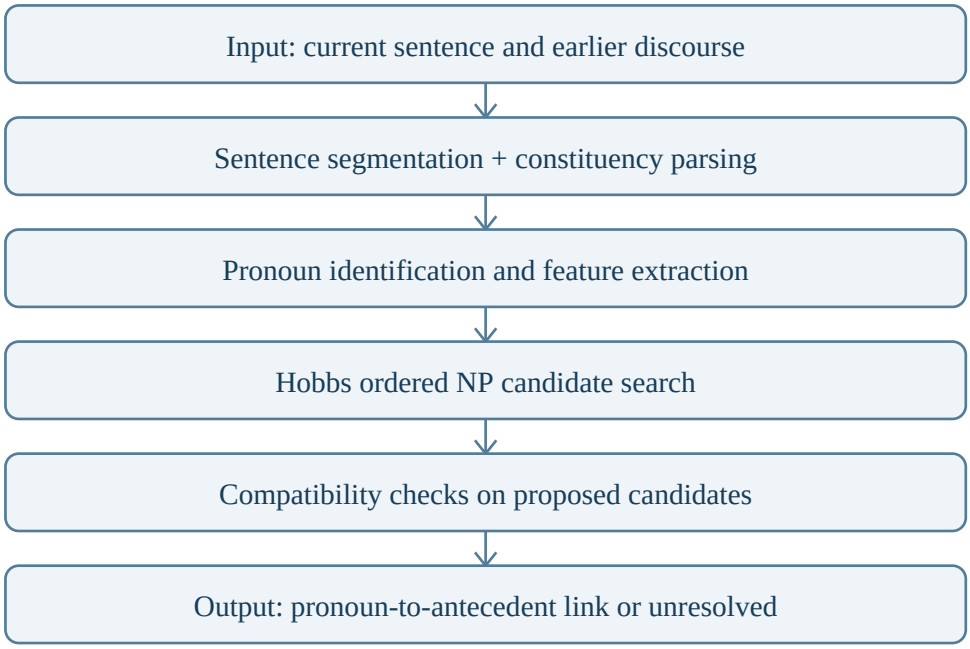
Scope of these notes

The focus is the classic syntax-based pronoun-search procedure. It is not a complete solution to every kind of anaphora. Reflexives, nonreferential “it”, split antecedents, and references requiring world knowledge need additional handling.

Learning goals

Understand how to climb a parse tree, identify branches relative to a path, use breadth-first traversal, reject incompatible candidates, and trace an antecedent decision.

3. Architecture



Architecture note: This is an illustrative implementation pipeline. A parser and a feature extractor are supporting components; the Hobbs search itself operates on existing parse trees.

Component	Role
Constituency parser	Builds phrase structure with parent-child relations and word order.
Pronoun identifier	Locates the target pronoun and its containing NP.
Feature extractor	Records number, person, and relevant gender / semantic information.
Search controller	Maintains the current node X, path p, candidate order, and earlier-sentence index.
Candidate checker	Accepts or rejects proposed NPs; unknown features need an explicit policy.
Resolution output	Returns a linked antecedent, a search trace, or an unresolved result.

Input and output example

Input: “The students arrived. They sat down.”
Target: They (third person, plural).
Output: They → the students. A program may store this as a link rather than replace the pronoun in the text.

4. Tree Concepts and Workflow

Symbol / concept	Meaning
NP	Noun phrase: “the teacher”, “a book”, or a pronoun phrase.
S	Sentence / clause node in the traditional tree notation.
VP	Verb phrase: “read a book”.
X	Current NP or S node used to control the search.
p	Path from X down to the previously selected lower node.
Dominates	A node dominates another node if that node lies in its subtree.
Immediately dominates	Direct parent-child relationship, with no intervening node.
Left / right of p	Sibling branches occurring before / after the path branch at each level.
Breadth-first	Visit shallower nodes before deeper ones; preserve left-to-right order within a level.

Step-by-step workflow

• **1. Prepare:** Parse the current sentence and retain earlier sentence trees.

• **2. Locate:** Find the NP directly containing the pronoun.

• **3. Search locally:** Climb to the first NP or S and examine eligible left-side branches.

• **4. Expand:** If necessary, climb to higher NP/S nodes and apply the corresponding search rules.

• **5. Move backward:** At the highest S, inspect preceding sentences from newest to oldest.

• **6. Filter:** Test every proposed NP in order; accept the first compatible one.

• **7. Finish:** If the permitted search space is exhausted, return unresolved.

Important distinction

“Nearest noun” means proximity in the word sequence. Hobbs uses structural search order. A shallow subject NP can be proposed before a deeper object NP even when the object is closer to the pronoun.

5. Algorithm: Nine Search Steps

The following is a compact paraphrase of the original search procedure [1]. At every proposal, apply compatibility checks; rejection means continuing the search.

Step	Action
1	Start at the NP immediately containing the pronoun.
2	Climb to the first NP or S. Set X to it; record the downward path p.
3	Search left of p breadth-first, left-to-right. Propose an NP only if another NP or S intervenes between it and X.
4	If X is the highest S, search earlier sentence trees, newest first, breadth-first left-to-right, proposing NPs. Otherwise continue.
5	Climb from X to the next NP or S; update X and p.
6	If X is NP, propose X unless p passes through its immediately dominated nominal head node (N-bar in the original notation).
7	Search branches below X left of p, breadth-first left-to-right; propose encountered NPs.
8	If X is S, search right of p breadth-first left-to-right. Propose NPs; do not descend below an encountered NP or S.
9	Return to step 4.

Compatibility checks: supporting policy

• **Number:** reject a known singular candidate for a plural pronoun in an ordinary agreement setting.

• **Gender / semantic suitability:** use known lexical or contextual information. Do not infer gender from a name alone.

• **Additional constraints:** person, animacy, binding, or nonreferential-pronoun handling can be included by an implementation. They are distinct from the search order.

Reading step 6: Trees differ in their treatment of N-bar and noun heads. A faithful implementation must map its parser’s labels to the intended nominal structure rather than simply ignore the condition.

6. Pseudocode and Search Logic

This teaching pseudocode keeps candidate generation separate from acceptance. HOBBS_CANDIDATES follows the nine steps on the previous page.

```
RESOLVE(pronoun, current_tree, previous_trees):
    if pronoun is known to be nonreferential:
        return NOT_APPLICABLE

    for candidate in HOBBS_CANDIDATES(
            pronoun, current_tree, previous_trees):
        if COMPATIBLE(pronoun, candidate):
            return candidate

    return UNRESOLVED

BREADTH_FIRST(ordered_roots, stop_rule):
    queue = ordered_roots in left-to-right order
    while queue is not empty:
        node = remove_front(queue)
        yield node
        if not stop_rule(node):
            append node.children left-to-right
```

How to configure breadth-first traversal

Search situation	Traversal behaviour
Initial local search	Restrict to left-side branches and apply the intervening NP/S test before proposing an NP.
Higher left-side search	Restrict to left-side branches; propose encountered NPs.
Higher right-side search	Stop descending at any encountered NP or S.
Earlier sentence search	Start at its root, search breadth-first left-to-right, and propose NPs.

Termination and cost

The search terminates on acceptance or exhaustion of the available trees. A practical system can limit discourse history, but that is an implementation choice. For V visited nodes and C proposed candidates, traversal and constant-time compatibility checks take roughly O(V + C). Repeated region searches, feature computation, and parsing add cost; this is not a universal complexity bound for all implementations.

7. Worked Example: Earlier Sentence

Discourse: “The students carried a bag. They rested.”
Target pronoun: They.
Assumption: Ordinary plural agreement; no special collective interpretation.

Simplified constituency trees

Sentence	Tree representation
Sentence 1	S → NP(The students) + VP(carried + NP(a bag))
Sentence 2	S → NP(They) + VP(rested)

Step-by-step trace

• **Step 1:** Begin at NP(They) in sentence 2.

• **Step 2:** Climb to S of sentence 2. This is X; p goes through its subject NP.

• **Step 3:** No eligible branches occur left of the subject path. No candidate is proposed.

• **Step 4:** X is the highest S. Move to sentence 1, the most recent earlier sentence.

• **Breadth-first visit:** S is visited first; then its NP and VP children. The subject NP(The students) is encountered before the deeper object NP(a bag).

• **Compatibility:** They is plural; the students is plural and refers to people. Accept it.

• **Output:** They → the students. The search stops before a bag is considered.

Candidate decision table

Candidate	Decision
The students	First proposed NP in the earlier tree; plural and compatible. Accepted.
A bag	Would occur later in the tree search. Not reached after acceptance; singular anyway.

Interpretation

The students carried a bag, and the same students rested. For classroom explanation, “They rested” can be expanded as “The students rested”. A production system can preserve the original wording and store the link.

8. Worked Example: Same Sentence

Sentence: “The teacher said that she was ready.”
Target: she.
Assumption: Context establishes that the teacher is a woman.

Simplified tree structure

Node	Children / content
S-main	NP(the teacher) + VP(said + S-embedded)
S-embedded	NP(she) + VP(was ready)

Step-by-step trace

• **Step 1:** Start at NP(she).

• **Step 2:** The first enclosing NP or S above it is S-embedded. Set X to that clause.

• **Step 3:** There is nothing eligible to the left of its subject path.

• **Step 4:** S-embedded is not the highest S; continue upward.

• **Step 5:** Climb through the main VP to S-main. Update X and p.

• **Step 6:** X is S, so the NP-specific test does not apply.

• **Step 7:** Search left of p. NP(the teacher), the main subject, is proposed.

• **Compatibility:** Singular number and the assumed contextual gender agree. Accept the teacher.

• **Output:** she → the teacher. Step 8 is not needed after acceptance.

Why the clause structure matters

The teacher belongs to the main clause, while she is the embedded clause subject. The algorithm finds the candidate after climbing to the main clause. This differs from simply choosing the noun immediately before the pronoun.

Initial-search restriction: a contrast

“The teacher praised her.” In a simple S → NP + VP tree, the subject NP has no intervening NP or S between itself and that S. The initial local search does not propose it as the antecedent of the ordinary object pronoun. If the teacher is intended to refer to the same person, standard English normally requires “herself”. Reflexive resolution needs its own handling.

9. Rejection and Ambiguity

Example A: reject by number

Discourse: “The teacher greeted the students. They smiled.”

Stage	What happens
Local search	No eligible local antecedent exists before the subject pronoun in the second sentence.
Earlier-tree candidate 1	The teacher is encountered at a shallower level. Singular: reject for plural They.
Earlier-tree candidate 2	The students is an object NP at a deeper level. Plural: accept.
Result	They → the students.

Example B: structurally preferred, not certain

Discourse: “Ravi met Arun. He smiled.”
Assume both named people are known male individuals. Both agree with He. In the simple parse, the earlier sentence subject Ravi is proposed before the object Arun. Hobbs selects Ravi, but the text itself can remain ambiguous.

Example C: common-sense challenge

“The trophy did not fit in the suitcase because it was too large.” Understanding “it” requires reasoning about size and fitting. Syntactic preference and agreement alone cannot reliably settle such examples. A semantic or contextual model may be needed.

Example D: no entity antecedent

“It is raining.” The it is expletive or nonreferential in this use. Forcing a link to a previous noun would be wrong. A supporting component should recognize this case.

Frequent mistakes

• Using depth-first search in place of breadth-first search.

• Treating the nearest noun as automatically correct.

• Omitting the initial intervening NP/S condition.

• Descending below NP/S nodes during the restricted right-side search.

• Searching earlier sentences in oldest-first order.

• Reporting an accepted candidate as certain when the passage is ambiguous.

10. Applications and Limitations

The following are illustrative uses of pronoun resolution in NLP. Hobbs can serve as a transparent baseline or component; these examples do not claim that modern systems necessarily use it.

Application	How resolution helps
Reading comprehension	“The students arrived. They sat.” links the action to the students.
Question answering	Links “She won the award” to an earlier named person when answering who won.
Information extraction	Associates a pronoun’s action with the correct entity record.
Summarization	Helps maintain entity continuity and avoid unclear pronouns in a summary.
Machine translation	Antecedent information can help choose suitable pronoun forms in a target language.
Dialogue processing	Connects “it” in a follow-up request to a previously mentioned item when context allows.
Teaching and evaluation	Provides an explainable search trace and a classical comparison baseline.

Advantages

• Deterministic and easy to inspect with a correct parse.

• Does not need labelled training examples for candidate ordering.

• Demonstrates how syntax affects pronoun interpretation.

Limitations

• Incorrect or incompatible parse structures change the candidate order.

• Agreement does not resolve all ambiguous cases.

• World knowledge, discourse focus, and speaker intention are not captured fully.

• Unknown features, singular they, collective nouns, and coordinated antecedents require careful policies.

• Classic English-oriented tree rules do not transfer unchanged to every language or parser.

Practical evaluation

Use passages with known antecedents and measure correct links among evaluated pronouns. Also inspect unresolved cases, parse failures, and ambiguous examples separately. Compare systems on the same data and assumptions; no universal accuracy value applies.

11. Revision and References

Short exam answer

Hobbs' algorithm resolves third-person pronouns using constituency-tree search. It begins at the pronoun NP, climbs through NP and S nodes, and proposes eligible NPs in a constrained breadth-first, left-to-right order. If the current sentence is exhausted, it searches earlier sentences from most recent to older. The first compatible candidate is selected; otherwise the pronoun remains unresolved.

Practice questions

- 1. Define anaphora, pronoun, and antecedent.
- 2. Why does breadth-first order differ from choosing the nearest noun?
- 3. Trace "The nurses helped a child. They returned."
- 4. Trace "A doctor greeted the nurses. They returned."
- 5. Why can "Ravi met Arun. He laughed" remain ambiguous?

Answer hints

1. Anaphora is backward-dependent reference; a pronoun is a referring grammatical form; its antecedent is the expression used to interpret it. **2.** Tree depth and left-to-right structure determine priority, not token distance. **3.** The earlier subject the nurses is plural and accepted. **4.** A doctor is singular and rejected; the nurses is plural and accepted. **5.** Both candidates may satisfy agreement; structural preference does not prove the intended meaning.

Reference and source note

[1] Hobbs, J. R. (1978). *Resolving pronoun references*. Lingua, 44(4), 311-338. DOI: 10.1016/0024-3841(78)90006-2.

Author-hosted scan: Resolving Pronoun References, USC Information Sciences Institute. The scan is a reprint with different page numbering. The nine-step procedure is on the page labelled 341.

The algorithm steps are paraphrased. Architectures, simplified parse trees, examples, pseudocode wrapper, and application scenarios in these notes are teaching explanations. Parser label conventions and compatibility policies must be stated in an implementation.