Discourse and Coreference

LING 571 — Deep Processing Methods in NLP
November 20, 2019
Shane Steinert-Threlkeld
Clarification

- In pseudocode from Monday:
  - Incrementing support is done *after* determination of MI-LCS.
  - In other words: each probe *word* only increments support for one sense of the target word.
Alternative Resnik WSD Pseudocode

Given: input word \( w_0 \) and probe words \( \{p_1, \ldots, p_n\} \)

for \( p_i \) in \( \{p_1, \ldots, p_n\} \):
    supported_sense = null
    most_information = 0.0
    for \( \text{sense}_w \) in \( \text{SENSES}(w_0) \):
        for \( \text{sense}_p \) in \( \text{SENSES}(p_i) \):
            \( \text{lcs}_{\text{synset}} = \text{LOWESTCOMMONSUBSUMER}(\text{sense}_w, \text{sense}_p) \)
            \( \text{lcs}_{\text{info}} = \text{INFORMATIONCONTENT}(\text{lcs}_{\text{synset}}) \)
            if \( \text{lcs}_{\text{info}} > \text{most}_{\text{information}} \):
                \( \text{most}_{\text{information}} = \text{lcs}_{\text{info}} \)
                supported_sense = \( \text{sense}_w \)
                increment support[\( \text{supported}_\text{sense} \)] by \( \text{most}_{\text{information}} \)
Roadmap

- Introduction to Discourse
- Coreference Resolution
  - Phenomena
  - Pronominal Anaphora Resolution
  - Hobbs’ Algorithm
Introduction to Discourse
What is Discourse?

- Discourse is “a **coherent structured group of sentences.**” (J&M p. 681)

- Discourse is language *in situ*
  - rather than synthetic, isolated sentences.
  - language use *toward a goal*
Different Parameters of Discourse

- **Number of participants**
  - Single author/voice → Monologue
  - Multiple participants → Dialogue

- **Modality**
  - Spoken vs. Written

- **Goals**
  - Transactional (message passing) vs. Interactional (relations, attitudes)
  - Cooperative task-oriented rational interaction
Why Discourse?

- Understanding depends on context
  - Word sense — *plant*
  - Intention — *Do you have the time?*
  - Referring expressions — *it, that, the screen*
  - Domain restriction — “All of the students read the announcement.”
Why Discourse?

- Applications: Discourse in NLP
  - Question-Answering
  - Information Retrieval
  - Summarization
  - Dialogue / Conversational AI
  - Automatic Essay Grading
Reference Resolution

- Knowledge sources:
  - *Domain Knowledge*
  - *Discourse Knowledge*
  - *World Knowledge*

**User:** Where is *A Bug’s Life* playing in *Summit*?
**System:** A Bug’s Life is playing at the Summit Theater.
**User:** When is *it* playing *there*?
**System:** It’s playing at 2PM, 5PM, and 8PM.
**User:** I’d like 1 *adult* and 2 *children* for the first show. How much would *that* cost?

*From Carpenter and Chu-Carroll, Tutorial on Spoken Dialogue Systems, ACL ‘99*
Not All Sentences Are Created Equal

- First Union Corp. is continuing to wrestle with severe problems.[1] According to industry insiders at PW, their president, John R. Georgius, is planning to announce his retirement tomorrow.[2]

Summary:

- First Union President John R. Georgius is planning to announce his retirement tomorrow.

Inter-sentence coherence relations:

- Second sentence: main concept (nucleus)
- First sentence: background
Coherence Relations

John hid Bill’s car keys. He was drunk.
John hid Bill’s car keys. He likes spinach.

● Why is this odd?
  ● No obvious relation between sentences
  ● Breaks our assumption as readers that information presented in discourse is relevant

● How is the first pair related?
  ● statement — explanation/cause

● Assumption: utterances should have meaningful connection
  ● Establish through coherence relations
Coherence Relations

John hid Bill’s car keys. He was drunk. John hid Bill’s car keys. He likes spinach.

● Assumption

● Segments of discourse should have meaningful connection.

● Establish through coherence relations
Discourse: Looking Ahead

Coreference
Cohesion
Coherence
Structure / Segmentation
Coreference Resolution
Reference: Terminology

- **referring expression** (refexp)
  - An expression that picks out entity (referent) in some knowledge model
  - Referring expressions used for the same entity **corefer**
    - Queen Elizabeth, her, the Queen
    - Logue, a renowned speech therapist
  - Entities in **purple** do not corefer to anything.

Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch. Logue, a renowned speech therapist, was summoned to help the King overcome his speech impediment.
Antecedent:

- An expression that introduces an item to the discourse for other items to refer back to
- Queen Elizabeth… her

Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch. Logue, a renowned speech therapist, was summoned to help the King overcome his speech impediment.
Reference: Terminology

- **Anaphora**: An expression that refers back to a previously introduced entity.
  - **Cataphora**: Introduction of expression before referent:
    - “Even before she saw it, Dorothy had been thinking about…”

*Not all anaphora is referential! e.g. “No dancer hurt their knee.”*
Referring Expressions

- Many forms:
  - Queen Elizabeth
  - she/her
  - the Queen
  - HRM
  - the British Monarch
Referring Expressions

- **Queen Elizabeth** – *she/her* – **the Queen** – **HRM** – **the British Monarch**

- “Correct” form depends on discourse context
  - *she, her* presume prior mention or presence in the world
  - **the Queen** presumes an Anglocentric geopolitical discourse context generally or the UK (or British Commonwealth) specifically

  (…i.e. likely a different interpretation during a RPDR viewing party.)
Discourse Model

- Correct interpretation of reference requires **Discourse Model**
  - Entities referred to in the discourse
  - Relationships of these entities
- Need way to construct, update model
  - First mention of entity *evokes* entity *into* model
    - [“introduces a discourse referent (dref)”]
  - Subsequent mentions *access* entity *from* the model.
Reference and Model

Discourse Model

Evocation

“Jane”

corefer

“she”

Access
Reference Tasks

● **Coreference resolution:**
  ● Find all expressions referring to the same entity in a text.
  ● A set of coreferring expressions is a *coreference chain*.

● **Pronominal anaphora resolution:**
  ● Find antecedent for a single pronoun.
  ● Subtask of coreference resolution
Pronominal Anaphora Resolution
## Reference Phenomena

<table>
<thead>
<tr>
<th>Expression Type</th>
<th>Examples</th>
<th>Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indefinite NP</td>
<td>“a cat”, “some geese”</td>
<td>Introduces new entity to context</td>
</tr>
<tr>
<td>Definite NP</td>
<td>“the dog”</td>
<td>Refers to entity identifiable by hearer in context</td>
</tr>
<tr>
<td>Pronouns</td>
<td>“he,” “them,” “they”</td>
<td>Refers to entity, must be “salient”</td>
</tr>
<tr>
<td>Demonstratives</td>
<td>“this,” “that”</td>
<td>Refers to entity, sense of distance (literal/figurative)</td>
</tr>
<tr>
<td>Names</td>
<td>“Dr. Woodhouse,” “IBM”</td>
<td>New or old entities</td>
</tr>
</tbody>
</table>
Reference Phenomena: Activation/Salience

a) John went to Erin’s party, and parked next to a classic Ford Falcon.
b) He went inside and talked to Erin for more than an hour.
c) Erin told him that she recently got engaged.
d) ?? She also said that she bought it yesterday.
   e) She also said that she bought the Falcon yesterday.

- d) is problematic because the Falcon has lost its salience.
- e) is acceptable because the definite NP has a further range for salience.
Information Status

- Some expressions introduce **new** information (ex: indefinite NPs)
- Other expressions refer to previous referents (ex: Pronouns)
- “**Givenness hierarchy**” (Gundel et al. 1993)

<table>
<thead>
<tr>
<th>in focus &gt;</th>
<th>activated &gt;</th>
<th>familiar &gt;</th>
<th>uniquely identifiable &gt;</th>
<th>referential &gt;</th>
<th>type identifiable</th>
</tr>
</thead>
<tbody>
<tr>
<td>it</td>
<td>this</td>
<td>that N</td>
<td>the N</td>
<td>indef. this N</td>
<td>a N</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Information Status

- **Accessibility scale**: (Ariel, 2001)
  - More salient elements easier to call up, can be shorter
  - correlates with length: more accessible, shorter refexp
Complicating Factors

- **Inferrables**
  - refexp refers to inferentially related entity:
    - *I bought a car today, but a door had a dent, and the engine was noisy.*
    - *a door, the engine ∈ a car*

- **Generics:**
  - *I want to buy a Jaguar. They are very stylish.*
  - General group evoked by instance.

- **Non-referential cases:**
  - *It’s raining.* (Pleonasm)
  - *It was good that Frodo carried the ring.* (Extraposition)
Features for Anaphora Resolution: Constraints

- **Number:**
  - *Anjali has a Corvette.*  *They are red.*  *It is red.*

- **Person:**
  - 1\textsuperscript{st}: I, we  2\textsuperscript{nd}: you, y'all  3\textsuperscript{rd}: he, she, it, they

- **Gender:**
  - *Janae plays the guitar.*  *She sounds great.*
  - *Janae plays the guitar.*  *It sounds great.*
Features for Anaphora Resolution: Constraints

- **Binding Theory**
  - How to handle reflexive pronouns vs. nonreflexives
  - *Aaron bought themself a new car.*
  - *Aaron bought them a new car.* [them ≠ Aaron]
  - *Jen said that Imani bought herself a new car.* [herself = Imani]
  - *Jen said that Imani bought her a new car.* [her ≠ Imani]
  - *He₁ said that he₂ bought Willie a new car.* [He₁ ≠ Willie, he₂ ≠ Willie]

- Pronoun/Def. NP: can’t corefer with subject of clause
- Reflexives do corefer with subject of containing clause
Features for Anaphora Resolution: Preferences

- **Recency:**
  - Prefer closer antecedents.
  - *The doctor found an old map in the captain’s chest. Jim found an even older map on the shelf. It described an island.*

- **Grammatical role:**
  - Saliency hierarchy of roles
  - e.g. \textit{Subj} > \textit{Object} > \textit{Ind. Object} > \textit{Oblique} > \textit{AdvP}
  - *Billy Bones went to the bar with Jim Hawkins.* He called for a glass of rum.
  - *Jim Hawkins went to the bar with Billy Bones.* He called for a glass of rum.
Features for Anaphora Resolution: Preferences

- **Repeated Mention:**
  - Once entity is focused, likely to continue to be focused \(\rightarrow\) more likely pronomialized.
  - *Billy Bones* had been thinking of a glass of rum. *He* hobbled over to the bar. *Jim Hawkins* went with him. *He* called for a glass of rum.

- **Parallelism:**
  - Prefer entity in same role.
  - *Silver* went with *Jim* to the bar. *Billy Bones* went with *him* to the inn.
Features for Anaphora Resolution: Preferences

● Verb Semantics

● Some verbs semantically bias for one of their argument positions.

  John telephoned Bill. He had lost the laptop.
  John criticized Bill. He had lost the laptop.

● Selectional Restrictions

● Other kinds of semantic knowledge

  John parked his car in the garage after driving it around for hours.
  ● Understood that a car has the ability to drive whereas garage does not.
Reference Resolution Approaches

● Common features:
  ● Use of a “Discourse Model”
    ● Referents evoked in discourse, available for reference
    ● Structure indicating relative salience
  ● Syntactic & Semantic Constraints
  ● Syntactic & Semantic Preferences

● Differences:
  ● Which constraints/preferences? How to combine? Rank?
Hobbs’ Algorithm
Hobbs’ Resolution Algorithm

● **Requires:**
  ● Syntactic parser
  ● Gender & number checker

● **Input:**
  ● Pronoun
  ● Parse of current and previous sentences

● **Captures:**
  ● Preferences: Recency, grammatical role
  ● Constraints: binding theory, gender, person, number
Hobbs Algorithm

● Summary:
  ● English-centric, rule-based algorithm.
  ● Exploits English features of:
    ● Agreement
    ● Right-branching
    ● SOV order
  ● Inter-sententially, exploits notions of recency.
Hobbs Algorithm Detail (Hobbs, 1978)

1. Begin at the noun phrase (NP) node immediately dominating the pronoun

2. Go up the tree to the first NP or sentence (S) node encountered. Call this node \(X\), and call the path used to reach it \(p\).

3. Traverse all branches below node \(X\) to the left of path \(p\) in a left-to-right, breadth-first fashion. Propose as the antecedent any encountered NP node that has an NP or S node between it and \(X\).

4. If node \(X\) is the highest S node in the sentence, traverse the surface parse trees of previous sentences in the text in order of recency, the most recent first; each tree is traversed in a left-to-right, breadth-first manner, and when an NP node is encountered, it is proposed as antecedent. If \(X\) is not the highest S node in the sentence, continue to step 5.
Hobbs Algorithm Detail \textsuperscript{(Hobbs, 1978)}

5. From node $X$, go up the tree to the first NP or S node encountered. Call this new node $X$, and call the path traversed to reach it $p$.

6. If $X$ is an NP node and if the path $p$ to $X$ did not pass through the Nominal node that $X$ immediately dominates, propose $X$ as the antecedent.

7. Traverse all branches below node $X$ to the \textit{left} of path $p$ in a left-to-right, breadth-first manner. Propose any NP node encountered as the antecedent.

8. If $X$ is an S node, traverse all branches of node $X$ to the \textit{right} of path $p$ in a left-to-right, breadth-first manner, but do not go below any NP or S node encountered. Propose any NP node encountered as the antecedent.

Hobbs Example

Lyn’s mom is a gardener.

Craige likes her.

Diagram:

- **S1**: NP1 → Det NP2 → ‘s mom → V NP → is Det → N3 → gardener
- **S2**: NP4 → Craige V NP5 → likes her
Hobbs Example

Lyn’s mom is a gardener.

Craige likes her.

Diagram of the sentence structure:

- **NP** refers to noun phrase.
- **VP** refers to verb phrase.
- **Det** refers to determiner.
- **N** refers to noun.
- **V** refers to verb.

**NP$_1$**
- Det (Lyn’s)
- N (mom)
- V (is)
- NP$_2$ (her)

**NP$_2$**
- Det (a)
- N (gardener)

**VP**
- N (is)

**NP$_3$**
- N (her)

**NP$_4$**
- N (Craige)

**NP$_5$**
- V (likes)
Lyn’s mom is a gardener. Craige likes her.

Hobbs Example
Hobbs Example

Lyn’s mom is a gardener. Craige likes her.
…the castle in Camelot remained the residence of the king until 536 when he moved it to London.

for full walkthrough see Hobbs, 1978 p. 317, and the end of today's slides
Hobbs Algorithm

● Results: 88% Accuracy; 90% intrasentential
  ● …on perfect, manually parsed sentences

● Useful baseline for evaluating pronomial anaphora

● Issues:
  ● Parsing:
    ● Not all languages have parsers
    ● Parsers not always accurate
  ● Constraints/Preferences:
    ● Captures: Binding theory, grammatical role, recency
    ● But not: parallelism, repetition, verb semantics, selection
Hobbs Algorithm

- Other issue: does not implement world knowledge

  - The city council refused the women a permit because they feared violence.
  
  - The city council refused the women a permit because they advocated violence. 
    
  (Winograd, 1972)*
  
  *more on this later

- Get this reading by knowledge of city councils and permitting, and reasons why permits would be refused.
Hobbs Algorithm: A Parable

- Was actually one of the first instances in NLP where a researcher tried an informed, if “naïve” baseline
- …found that (in 1972) no system he could build could beat it!

- “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.” — Hobbs (1978), Lingua, p. 345
Coreference and World Knowledge

- The trophy doesn't fit into the brown suitcase because it's too [small/large]. What is too [small/large]?  
  - Answers: The suitcase/the trophy.

- Joan made sure to thank Susan for all the help she had [given/received]. Who had [given/received] help?  
  - Answers: Susan/Joan.

- Paul tried to call George on the phone, but he wasn't [successful/available]. Who was not [successful/available]?  
  - Answers: Paul/George.

- The lawyer asked the witness a question, but he was reluctant to [answer/repeat] it. Who was reluctant to [answer/repeat] the question?  
  - Answers: The witness/the lawyer.
Winograd Schema Challenge

- Still hard!
- **WSC**

Heavily supervised

---

<table>
<thead>
<tr>
<th>Rank</th>
<th>Name</th>
<th>Model</th>
<th>URL</th>
<th>Score</th>
<th>BoolQ</th>
<th>CB</th>
<th>COPA</th>
<th>MultiRC</th>
<th>ReCoRd</th>
<th>RTE</th>
<th>WIC</th>
<th>Score</th>
<th>WSC</th>
<th>AX-g</th>
<th>AX-b</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SuperGLUE Human Baselines</td>
<td>SuperGLUE Human Baselines</td>
<td>🔄</td>
<td>89.8</td>
<td>89.0</td>
<td>95.8/98.9</td>
<td>100.0</td>
<td>81.8/51.9</td>
<td>91.7/91.3</td>
<td>93.6</td>
<td>80.0</td>
<td>100.0</td>
<td>99.3/99.7</td>
<td>76.6</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>T5 Team - Google</td>
<td>T5</td>
<td>🔄</td>
<td>88.9</td>
<td>91.0</td>
<td>93.0/96.4</td>
<td>94.8</td>
<td>86.2/62.3</td>
<td>93.3/92.5</td>
<td>92.5</td>
<td>78.1</td>
<td>93.8</td>
<td>92.7/91.9</td>
<td>65.6</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Facebook AI</td>
<td>RoBERTa</td>
<td>🔄</td>
<td>84.6</td>
<td>87.1</td>
<td>90.5/95.2</td>
<td>90.8</td>
<td>84.4/52.5</td>
<td>90.6/90.0</td>
<td>86.2</td>
<td>69.9</td>
<td>89.0</td>
<td>91.0/78.1</td>
<td>57.9</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>IBM Research AI</td>
<td>BERT-mtl</td>
<td>🔄</td>
<td>73.5</td>
<td>84.8</td>
<td>89.6/94.0</td>
<td>73.8</td>
<td>73.2/30.5</td>
<td>74.6/74.0</td>
<td>84.1</td>
<td>66.2</td>
<td>61.0</td>
<td>97.8/57.3</td>
<td>29.6</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>SuperGLUE Baselines</td>
<td>BERT++</td>
<td>🔄</td>
<td>71.5</td>
<td>79.0</td>
<td>84.8/90.4</td>
<td>73.8</td>
<td>70.0/24.1</td>
<td>72.0/71.3</td>
<td>79.0</td>
<td>69.6</td>
<td>64.4</td>
<td>99.4/51.4</td>
<td>38.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>BERT</td>
<td></td>
<td>🔄</td>
<td>69.0</td>
<td>77.4</td>
<td>75.7/83.6</td>
<td>70.8</td>
<td>70.0/24.1</td>
<td>72.0/71.3</td>
<td>71.7</td>
<td>69.6</td>
<td>64.4</td>
<td>97.8/51.7</td>
<td>23.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Most Frequent Class</td>
<td></td>
<td>🔄</td>
<td>47.1</td>
<td>62.3</td>
<td>21.7/48.4</td>
<td>50.0</td>
<td>61.1/0.3</td>
<td>33.4/32.5</td>
<td>50.3</td>
<td>50.0</td>
<td>65.11</td>
<td>30.0/50.0</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CiteW</td>
<td></td>
<td>🔄</td>
<td>44.5</td>
<td>62.2</td>
<td>49.0/71.2</td>
<td>51.8</td>
<td>0.0/0.5</td>
<td>14.0/13.6</td>
<td>49.7</td>
<td>53.1</td>
<td>65.11</td>
<td>30.0/50.0</td>
<td>-0.4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Outside Best</td>
<td></td>
<td>🔄</td>
<td>-</td>
<td>80.4</td>
<td>-</td>
<td>84.4</td>
<td>70.4/24.5</td>
<td>74.8/73.0</td>
<td>82.7</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Stanford Hazy Research</td>
<td>Snorkel (SuperGLUE v1.9)</td>
<td>🔄</td>
<td>-</td>
<td>-</td>
<td>88.6/93.2</td>
<td>76.2</td>
<td>76.4/03.6</td>
<td>78.9</td>
<td>72.1</td>
<td>72.6</td>
<td>47.6</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>
HW #9
Goals

- Explore the task of pronomial anaphora resolution
- Gain familiarity with syntax-based resolution techniques
- Analyze the effectiveness of the Hobbs algorithm by applying it to pairs of parsed sentences.
Task

- Given pairs of sentences \((S_0, S_1)\) as context
  - Resolve the pronoun(s) in \(S_1\) using the Hobbs algorithm.
  - J&M p. 704-705

**Subtasks:**
- Parsing Sentences — Automatic (CKY, Earley, etc)
- Hobbs Algorithm — May be done either:
  - Manually — manually mark up the output parse tree
  - Coded — implement Hobbs algorithm — will require feature grammar or similar for finding agreement, etc.
Notes

● For implementation

  ● May use any NLTK tools for parse tree manipulation

  ● …as long as it doesn’t directly implement the Hobbs algorithm!

● May create lookup table/dictionary for agreement

● Two results files:

  ● One for all parsed output

  ● One for remaining manual steps

  ● (Based on a copy of the first)
NLTK Tools

● “Climbing” parse trees:
  ● NLTK ParentedTree
    ● nltk.org/howto/tree.html
  ● Conversion from standard tree $t$
    ● $\text{parented}\_\text{tree} = \text{nltk.tree.ParentedTree.convert}(t)$

● Accessing feature structures
  $f_s = \text{nltk.grammarFeatStructNonterminal(parented_tree.label())}$
  $\text{pronoun}\_\text{agr} = f_s[\text{‘agr’}]$
  $\text{antecedent}\_\text{agr}.\text{subsumes(pronoun}\_\text{agr})$
Hobbs Algorithm Walkthrough
(h/t Ryan Georgi)
Let’s figure out what the antecedent for “it” is

(Hobbs, 1978 p. 318)
1. Begin at the noun phrase (NP) node immediately dominating the pronoun

(Hobbs, 1978 p. 318)
2. Go up the tree to the first NP or sentence (S) node encountered. Call this node $X$, and call the path used to reach it $p$.

(Hobbs, 1978 p. 318)
3. Traverse all branches below node $X$ to the left of path $p$ in a left-to-right, breadth-first fashion. Propose as the antecedent any encountered NP node that has an NP or S node between it and $X$.

(Quoted text from Hobbs, 1978 p. 318)
4. If node $X$ is the highest $S$ node in the sentence...

5. From node $X$, go up the tree to the first $NP$ or $S$ node encountered. Call this new node $X$, and call the path traversed to reach it $p$.

(Hobbs, 1978 p. 318)
6. If $X$ is an NP node and if the path $p$ to $X$ did not pass through the Nominal node that $X$ immediately dominates, propose $X$ as the antecedent.

“536” can’t be “moved”!

(Hobbs, 1978 p. 318)
7. Traverse all branches below node $X$ to the left of path $p$ in a left-to-right, breadth-first manner. Propose any NP node encountered as the antecedent.

(Hobbs, 1978 p. 318)
8. If \( X \) is an \( S \) node…


4. If node \( X \) is the highest \( S \) node in the sentence…

5. From node \( X \), go up the tree to the first \( NP \) or \( S \) node encountered. Call this new node \( X \), and call the path traversed to reach it \( p \).

\[(\text{Hobbs, 1978 p.318})\]
If $X$ is an NP node...

(Hobbs, 1978 p. 318)
7. Traverse all branches below node $X$ to the left of path $p$ in a left-to-right, breadth-first manner. Propose any NP node encountered as the antecedent.

(Hobbs, 1978 p. 318)
7. Traverse all branches below node $X$ to the left of path $p$ in a left-to-right, breadth-first manner. Propose any NP node encountered as the antecedent.

(Moving castles? 😐)

(Hobbs, 1978 p. 318)
7. Traverse all branches below node $X$ to the *left* of path $p$ in a left-to-right, breadth-first manner. Propose any NP node encountered as the antecedent.

```
the castle in
```

```
NP3
remained
```

```
the residence
```

```
Camelot
```

```
of NP6 536 when S
```

```
the king
```

```
he moved
```

```
NP
```

```
VP
```

```
VP
```

```
it to NP
```

```
London
```

```

“the residence of the king”
```

(Hobbs, 1978 p. 318)
Hobbs Algorithm Detail (Hobbs, 1978)

1. Begin at the noun phrase (NP) node immediately dominating the pronoun.

2. Go up the tree to the first NP or sentence (S) node encountered. Call this node \( X \), and call the path used to reach it \( p \).

3. Traverse all branches below node \( X \) to the left of path \( p \) in a left-to-right, breadth-first fashion. Propose as the antecedent any encountered NP node that has an NP or S node between it and \( X \).

4. If node \( X \) is the highest S node in the sentence, traverse the surface parse trees of previous sentences in the text in order of recency, the most recent first; each tree is traversed in a left-to-right, breadth-first manner, and when an NP node is encountered, it is proposed as antecedent. If \( X \) is not the highest S node in the sentence, continue to step 5.
Hobbs Algorithm Detail (Hobbs, 1978)

5. From node $X$, go up the tree to the first NP or S node encountered. Call this new node $X$, and call the path traversed to reach it $p$.

6. If $X$ is an NP node and if the path $p$ to $X$ did not pass through the Nominal node that $X$ immediately dominates, propose $X$ as the antecedent.

7. Traverse all branches below node $X$ to the left of path $p$ in a left-to-right, breadth-first manner. Propose any NP node encountered as the antecedent.

8. If $X$ is an S node, traverse all branches of node $X$ to the right of path $p$ in a left-to-right, breadth-first manner, but do not go below any NP or S node encountered. Propose any NP node encountered as the antecedent.

Hobbs Example

Lyn’s mom is a gardener. Craige likes her.

I. Begin at the noun phrase (NP) node immediately dominating the pronoun
Hobbs Example

*Lyn’s mom is a gardener.*

*Craige likes her.*

2. Go up the tree to the first NP or sentence (S) node encountered. Call this node \( X \), and call the path used to reach it \( p \).
Hobbs Example

Lyn’s mom is a gardener. Craige likes her.

3. Traverse all branches below node $X$ to the left of path $p$ in a left-to-right, breadth-first fashion. Propose as the antecedent any encountered NP node that has an NP or S node between it and $X$. 
Lyn’s mom is a gardener. Craige likes her.

4. If node $X$ is the highest $S$ node in the sentence, traverse the surface parse trees of previous sentences in the text in order of recency, the most recent first; each tree is traversed in a left-to-right, breadth-first manner, and when an NP node is encountered, it is proposed as antecedent.
Hobbs Example

- What about…?
  - Lyn’s mom is hired a gardener.
  - Craige likes her.
Coherence Relations

● **Elaboration**: Infer the same proposition $P$ from the assertions of $S_0$ and $S_1$.

$\bullet$ *Dorothy was from Kansas. She lived in the midst of the great Kansas prairies.*

● **Occasion**: A change of state can be inferred from the assertion of $S_0$ whose final state can be inferred from $S_1$, or a change of state can be inferred from the assertion of $S_1$.

$\bullet$ *Dorothy picked up the oil-can. She oiled the Tin Woodman’s joints.*
Coherence Relation Hierarchy

S1 – Armin went to the bank to deposit his paycheck
S2 – He then took a train to Kim’s car dealership.
S3 – He needed to buy a car.
S4 – The company he works for now isn’t near any public transportation.
S5 – He also wanted to talk to Bill about their softball league.
S1 – Armin went to the bank to deposit his paycheck
S2 – He then took a train to Kim’s car dealership.
S3 – He needed to buy a car.
S4 – The company he works for now isn’t near any public transportation.
S5 – He also wanted to talk to Bill about their softball league.

Coherence Relation Hierarchy

Occasion \((e_1; e_2)\)
- S1 \((e_1)\)
- Explanation \((e_2)\)
  - S2 \((e_2)\)
  - Parallel \((e_3; e_5)\)
    - Explanation \((e_3)\)
    - S5 \((e_5)\)
    - S3 \((e_3)\)
    - S4 \((e_4)\)
Coherence Relation Hierarchy

S1 – Armin went to the bank to deposit his paycheck.

S2 – He then took a train to Kim’s car dealership.

S3 – He needed to buy a car.

S4 – The company he works for now isn’t near any public transportation.

S5 – He also wanted to talk to Bill about their softball league.
S1 – Armin went to the bank to deposit his paycheck
S2 – He then took a train to Kim’s car dealership.
S3 – He needed to buy a car.
S4 – The company he works for now isn’t near any public transportation.
S5 – He also wanted to talk to Bill about their softball league.