November 13, 2003

Chapter 18.1

Reference Resolution

Overview

- Discourse
- Referring expressions
- Choosing an expression/recovering an antecedent
- Constraints
- Algorithms

Discourse

- What is discourse? (in the linguistic sense)
- What types of discourse are there?
- What makes discourse structured?
- Why is discourse relevant to computational linguistics?

Reference

- Speakers refer to entities with referring expressions.
  - reference
  - coreference
  - referent
  - antecedent
  - anaphora
  - cataphora

Referring expressions

- indefinite noun phrases
- definite noun phrases
  - definite descriptions
  - proper names
- pronouns
  - reflexive v. non-reflexive
  - bound pronouns
- demonstratives
- *one*-anaphora
- null pronouns (“pro-drop”)

Referents: Complications

- inferrables

  In a classroom, *the teacher* usually stands at the front.

- discontinuous sets

  Kim met Sandy at the store. *They* went out for coffee.

- generics

  Toby likes the siamangs at the zoo. *They* are a kind of monkey.

Three views on reference

- Competence: What kinds of referring expressions exist, and which kinds of referents can each be used to refer to? What constitutes *felicitous* use of a referring expression?
- Performance:
  - Production: How do speakers choose which kind of referring expression to use, for a given referent in a given context?
  - Comprehension: Given a referring expression, how do/can hearers recover the intended referent?

Two kinds of constraints

- Categorical (or near categorical): can be used to determine a set of candidates
- Soft or non-categorical: used to select among the set of candidates (if there's more than one)
- Constraints may conflict

(Categorial) constraints on antecedent selection (for pronouns)

- Agreement: Number, person, noun class (gender)
  - Examples from English?
  - Examples from other languages?
- Syntactic constraints on reflexive v. non-reflexive pronouns
- Selectional restrictions
  - Examples?

Soft constraints on antecedent selection

- Salience:
  - Recency
  - Grammatical role
  - Repeated mention
  - Verb semantics
- Parallelism

Three algorithms for pronoun resolution

- Salience factors algorithm (Lappin and Leass 1994)
- Tree search algorithm (Hobbs 1978)
- Centering algorithm (Brennan et al 1987, there are others)

Salience factors

- Uses a *discourse model* to keep track of referents
- Referents are each associated with a *salience value*
- Salience values are calculated based on mentions of the referent in the text

Updating the discourse model

- For each referring expression (as its referent is known), compute a salience value, from the following (determined empirically):
  - Sentence recency: 100
  - Subject: 80
  - Existential: 70
  - Direct object: 50
  - Indirect object or oblique: 40
  - Non-adverbial (not in a demarcated PP): 50
  - Head noun: 80

Updating the discourse model

- Each time a new sentence is considered, divide all existing values by 2.
- Values for expressions referring to referents already in the model get added to the existing values (except in the case where multiple expressions occur in the same sentence).

Choosing a referent for a pronoun

- Collect the potential referents (up to four sentences back)
- Exclude those that don't agree in person or number, or don't pass intrasentential syntactic coreference constraints (hard constraints).
- Compute the total salience value for each referent by adding to the existing value:
  - 35, for syntactic parallelism
  - -175, for cataphora
- Pick the referent with the highest salience value, breaking ties by picking the closest one in the string.

Tree search

- Doesn’t have an explicit discourse model
- Doesn’t explicitly encode preferences
- Approximates some preferences by specifying the order in which to search
- Must specify a grammar to build the trees

Tree search

1. Begin at the NP immediately dominating the pronoun.
2. Go up the tree to the first NP or S encountered (call this X and the path to reach it p).
3. Traverse all branches of X to the left of p in left-to-right, breadth-first fashion. Propose as the antecedent any NP node that is encountered which has an NP or S node between it and X.

Tree search

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 an antecedent. If X is not the highest S node in the sentence, proceed to next step.

Tree search

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 p did not pass through the Nominal node immediately dominated by X, propose X as the antecedent.
7. Traverse all branches below node X to the *left* of p in a left-to-right, breadth-first manner. Propose any NP encountered as the antecedent.

Tree search

8. If X is an S node, traverse all branches of 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. (cataphora)
9. Go to Step 4.

Centering Theory

- Discourse model keeps track of entity being "centered" on in each utterance.
- Every utterance (U_n) has:
  - $C_b(U_n)$ the '*backward looking center*', which is the entity being centered on after U_n is interpreted.
  - (Exception: For the first utterance in a discourse is undefined.)
  - $C_f(U_n)$ the '*forward looking centers*', an ordered (e.g., by grammatical role) list of all the entities referred to in U_n.

Centering Theory

- Every utterance (U_n) has:
  - ...
  - $C_b(U_{n+1}) =$ the highest ranked element of $C_f(U_n)$ mentioned in U_{n+1} 'by definition'.
  - $C_p(U_n)$ the *preferred center*, the highest ranking element of $C_f(U_n)$.

Intersentential relationships

	$C_b(U_2) = C_b(U_1)$ or undefined $C_b(U_1)$	$C_b(U_2) \neq C_b(U_1)$
$C_b(U_2) = C_p(U_2)$	Continue	Smooth-Shift
$C_b(U_2) \neq C_p(U_2)$	Retain	Rough-Shift

Rules

- Rule 1: If any element of $C_f(U_n)$ is realized by a pronoun in utterance U_{n+1}, then $C_b(U_{n+1})$ must be too.
- Rule 2: Transition states are ordered:

  Continue $\rangle$ Retain $\rangle$ Smooth-Shift $\rangle$ Rough-Shift

Algorithm

- Generate possible C_b-C_f combinations for each possible set of reference assignments.
- Filter by constraints (e.g., syntactic coreference constraints, selectional restrictions, centering rules and constraints).
- Rank by transition orderings.
- Pick the reference assignment that gives the most preferred relation in Rule 2.

Overview

- Discourse
- Referring expressions
- Choosing an expression/recovering an antecedent
- Constraints
- Algorithms