Last class: Parsing

1. Bottom-up chart parsing
2. Earley algorithm

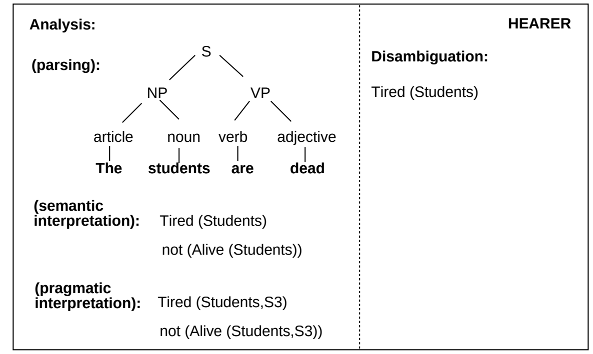
Today: Pragmatics and the problem of inference

- Text coherence
- Scripts
- Plan-based models of text understanding

Slide CS674–1

Pragmatics

Understanding sentences in context.



Slide CS674–2

Interpretation in Context

Jack took out a match. He lit a candle.

Jack took out a match. The sun set.

Useful to divide context into:

- **discourse context:** information from preceding text
- **situational context:** relevant world knowledge

Slide CS674–3

The Problem of Inference

When the balloon touched the light bulb, it broke. This made the baby cry. Mary gave John a dirty look and picked up the baby. John shrugged and picked up the balloon.

Slide CS674–4

NLU as Abduction

If A → B is true and B true, then A true.

X = *Fred desperately needed money for the mortgage payment.*

B = *Fred called his sister.*

Rule1 = If you need money then you can get it from a family member.
Rule2 = If you want to get something from someone, then you can ask them for it.
Rule3 = One way to ask someone for something is to call them.

Framework for Using World Knowledge

Expectation-Based Processing

1. Assume setting of discourse is represented by content of previous sentences and any inferences made when interpreting those sentences.
2. Use this information to **generate a set of expectations** about plausible eventualities.
3. **Match** possible interpretations of new sentences against expectations generated from the prevous discourse.

Knowledge About Action and Causality

Forms of Causality:

[**effect causality**] Set of intended effects or side effects typically caused by an action.

[**precondition causality**] Set of conditions that typically must hold just before action starts.

[**enablement**] A enables B if the effects of the first establish the preconditions of the second.

[**decomposition**] A is a substep of B if A is the first is one of a sequence of steps that constitute the execution of B.

Definition of BUY

Roles: Buyer, Seller, Object, Money

Constraints: Human(Buyer), SalesAgent(Seller), IsObject(Object), Value(Money, Price(Object))

Preconditions: AT(Buyer, Loc(Seller)), OWNS(Buyer, Money), OWNS(Seller, Object)

Effects: ¬OWNS(Buyer, Money), ¬OWNS(Seller, Object), OWNS(Buyer, Object), OWNS(Seller, Money)

Decomposition: GIVE(Buyer, Seller, Money), GIVE(Seller, Buyer, Object)

Scripts [Schank & Abelson]

- Prepackaged chain of causal relations between events and states that encodes expectations.
- Don't have to generate expectations from first principles using causality reasoning.
- Knowledge structure that encodes stereotypical sequences of events.

John was hungry. He went into Goldstein's and ordered a pastrami sandwich. It was served to him quickly. He left the server a large tip.

$RESTAURANT Script

Roles: Customer(S), Server(W), Cook(C), Cashier(M), Food(F)

Props: Table, Utensils, etc.

Constraints: HUMAN(S), HUMAN(W), etc.

Preconditions: HAS-MONEY(S)

Effects:
HAS-LESS-MONEY(S), HAS-MORE-MONEY(M), ¬HUNGRY(S),¬PLEASED(S)

Decomposition (Conceptual Dependency form):

1. **Enter:** S PTRANS S into Restaurant; S ATTEND Eyes to Tables; S MBUILD where to sit; S PTRANS S to Table; S MOVE S to sitting position.
2. **Order:** S MTRANS food-order to W (main)
3. **Eat:** S INGEST X (main)
4. **Exit:** S ATRANS money to M (main)

Using Scripts to Understand a Story

Assume: script $S, consisting of events e_1, e_2, ...
For each sentence, s in text:

1. Parse s into its propositional CD form.
2. While event, e, in list of script events:
   (a) If s matches e,
      i. Instantiate e with current script roles.
      ii. Instantiate all intervening events, i, with current script roles.
   (b) Else move pointer to next event, saving e in i.

Output is instantiated script.

Slide CS674–13

Problems with Scripts

1. Script selection
2. Managing multiple scripts
3. Aborting scripts
   John went to Goldstein's. He left.
4. Allowing for optional paths through scripts
   John was pick-pocketed on the way to restaurant.
5. Knowledge engineering requirements

Slide CS674–14

Novel Situations

John was hungry. He took out some ground beef.

John was hungry. He took out the Yellow Pages.

John needed money for the mortgage payment. He called his sister.

John needed money for the mortgage payment. He got a gun.

Slide CS674–15

Plan-Based Models of Text Understanding

A **plan** is a set of actions, that if executed would achieve some goal. A **goal** is a state that an agent wants to make true or an action that an agent wants to execute.

Task for NLP system: **plan recognition**

Input:

1. list of goals agents might be pursuing
2. set of actions that have been described or observed

Object: infer/track progress of plans of agents

Slide CS674–16

Decomposition Chaining

Top-down breadth-first search through *decomposition* links of action.

Sue wanted to take a train to Boston. She walked up to the ticket clerk.

Slide CS674–17

TRAVEL-BY-TRAIN

Roles: Actor, Clerk, SourceCity, DestCity, Train, Station, ...

Preconditions: OWNS(Actor, Money), At(Actor,SourceCity)

Effects: ¬OWNS(Actor, Money), At(Actor,DestCity), ...

Decomposition:

1. GoTo(Actor,Station)
2. Purchase-Ticket(Actor,Clerk,Ticket,Station)
3. GoTo(Actor,Loc(Train))
4. GetOn(Actor,Train)
5. Travel(Train, SourceCity, DestCity)

Slide CS674–18

6. Arrive(Train, DestCity)
7. GetOff(Actor,Train)

Slide CS674–19

Generalizing the Approach

James needed to be in Amherst by noon. He bought a ticket at the station.

1. Allow states to be goals as well as actions.
2. Plan recognition algorithm searches from actions to states via the actions' effects and vice versa.

Slide CS674–20

Enablement Causality

Jason bought new skates. He played in the ice hockey scrimmage Saturday night.

A Plan Recognition Algorithm

Interpret new sentence S:

[goal] 1. If P empty and S an *action*, add action to P.

2. If P empty and S an *state*, find all actions $A_1, \ldots, A_n$ that could have S as an effect. Create new plans P' for each.
3. If P not empty, then S may be a subgoal of P. Try to incorporate S into P.
4. If P not empty, then S may be a higher-level goal of P. Make S current goal; create new plan, P', and incorporate P into P'.

[action] 1. If P empty, incorporate as *goal*.

2. If P not empty and S matches an *action* in P, or one of its *effects* matches a *state* in P, or it has a *precondition* that matches a *state* in P, add S to P.
3. Otherwise, expand actions in P and try again.

[state] 1. If P empty, add to P.

2. If P not empty, attempt to match.
3. If P not empty and no match, add to P.

Limitations

- **No bottom-up component.**
  James bought a ticket to Amherst. He boarded the train at 6AM.
- **Inanimate causality.**
  Norma left some ice cubes in a bowl. Her mother was upset when she found that her antique earthen bowl was wet.
- **Auxiliary assumptions.**
  Richard needed some money. He went to ask his mother for some.

Richard needed some money. He wrote a proposal to the NSF.

- **Methods for determining preferred plans.**
  Fran picked up the newspaper...
- **Recognizing undesirable states as initiating goals.**
  Fred was hungry...
- **Handling obstacles to plans.**
  Fred was hungry. He went to ask his mother for a quarter to buy ice cream, but he couldn't find her. So he went to get an apple from the refrigerator. But the refrigerator was practically empty. To take his mind off

his growling stomach, he went out to play.

Beyond Plans

- affect-state maps
- plot units
- thematic action units
- . . .