Robust Parsing Seminar

Spring 1999

Instructors: Alon Lavie and Carolyn Rose

Robust Parsing Seminar

- **Time/Location:** Tuesdays, 3:00-5:00pm, LTI Red Room
- **Course Credit:** 9 units with project, 6 units without
- Sign up for 11-920 section-B (Independent Study - Breadth)
- **Student Responsibilities:**
  - Present 1-4 papers (in 1-2 class periods) and lead discussion.
  - Read assigned papers each week, write and submit a one page summary of the papers and main discussion points.
  - Optional limited project (or research paper)

Robust Parsing

The Problem:

- Conventional parsers are fragile
- Complete grammatical coverage is difficult to achieve
- Input in practical NL applications is unpredictable and difficult to control
- Major problem in analysis of unrestricted text
- Even greater problem in analysis of spoken input

Problems in Spoken Language Analysis

- Spoken language is **very** different from written text:
  - Disfluencies
  - Effects of interaction: back-channels, cut-off sentences
  - Differences in notions of grammaticality
  - Slurred speech
  - Lack of punctuation, clearly marked boundaries
  - Speech Recognition Errors

Parsing Spontaneous Speech

Disfluencies in Spontaneous Speech:

- False beginning, filled pauses:
  `um okay then yeah` **Monday at two is fine.**
- Repetition, filled pauses:
  `uh I` **I need to meet with you next week.**
- Incomplete sentences:
  `well,` **I'll be in New York next week** `and I was hoping`
  `<pause>`

Other Common Interpretation Problems

- Agreement errors:

  `two lightbulb, more than one sources`

- Deleted words and constituents:

  `A: when are you available?`

  `B: [I'm available on] Monday and Tuesday`

- Word substitutions, spelling errors: to/two, for/four

- Complex structures not covered by the grammar

Robust Parsing

Goals:

- Interpret underlying meaning of an input utterance
- Ignore disfluencies, non-important portions of input
- Augment input with “missing” information
- Graceful degradation in performance

Why is it hard?

- Correct interpretation of ill-formed input requires much knowledge
- With limited knowledge - involves a large search space

Robust Parsing

Main Issues:

- **Parser Flexibility:** what types of disfluencies and deviations from the grammar can the parser handle?
- **Parser Search:** What is the search space over which interpretations are sought? How is the space searched and with what efficiency? Pruning techniques?
- **Parser Efficiency**
- **Parse Selection and Disambiguation:** How are potential parses and sub-parses evaluated and ranked? How to cope with high levels of ambiguity?
- Where in the analysis process is robustness applied?
- What knowledge is applied in the interpretation process?

Taxonomy of Main Approaches to Robust Parsing

- Grammar and/or Domain-specific Repair Approaches
- Full Minimal Distance Parsers
- Limited Flexibility Approaches
- Statistical and Connectionist Parsers
- Multi Stage/Level Approaches

Grammar-specific Repair Heuristics

[Jensen, early ’90s], [Hobbs, 1991], [McDonald 1993]

- First Pass - attempt to parse in full with standard grammar
- If fails - “repair” the parse using grammar-specific rules
- Depends on bottom-up Chart Parser that will leave partial results at end of first pass
- Repair rules or heuristics specify how to combine partial parses, or what grammatical constraints to relax in second pass
- Repair heuristics are grammar-specific, not a general solution
- Not well suited for speech - degradation usually not graceful

Full Minimal Distance Parsers

[Lehman, 1989], [Smith and Hipp, 1992], [Ramshaw, 1994]

- Find a grammatical sentence of minimum distance to input and analyze it instead.
- Operations allowed usually include: insertions, deletions, substitutions and transpositions.
- Various heuristics can be used for calculating the “distance”:
  - Penalty schemes for various operations
  - Semantic importance of words can be taken into account
  - Statistical models
- Meaning can be interactively confirmed with user
- Adaptive learning - grammar can be modified/augmented incrementally
- Main problem: Full MDP infeasible with large practical NL grammars

Skipping and Maximal Coverage Parsers

Phoenix [Ward,1989], GLR* [Lavie,1995], SOUP [Gavalda, 1998]

- A limited version of MDP, allowing mainly deletions (possibly also substitutions)
- Search for a parse that obtains maximal coverage of input
- Particularly well suited to parsing speech - search for coherent concepts
- Where are deletions allowed? - anywhere? only between main concepts?
- Additional heuristics to deal with high levels of ambiguity and selection of the most appropriate analysis.
- A more general solution to the problem, but crucial heuristics are domain/grammar dependent
- Are not as flexible as full MDP parsers

Connectionist Parsers

[Jain, 1991], [Gorin, 1993], [Buo, 1996]

- Idea: use Neural Networks to learn mappings between strings and output structures
- Often combined with some symbolic computation to produce the partial parses or the training corpus
- No need for explicit grammar development - grammar is “learned” from examples
- Very robust and graceful degradation
- Works well for rather shallow representations, difficult to develop for deeper symbolic representations.

Automatic and Interactive Repair Approaches

[Rose, 1997], [Rose and Waibel, 1997]

- Main Idea: a two stage process:
  - First do partial parsing (non robust or limited)
  - Then do a repair stage using domain (semantic) knowledge
  - Possibly confirm repair steps or selection of output via user interaction
- Better division of labor resources between stages
- Rose uses a Genetic Programming approach to search the repair space - construct a set of good output representation hypotheses
- Interaction with the user can narrow down on the correct/best hypothesis
- Equivalent in power to MDP, but more efficient and can work for large practical grammars and applications