Computational Psycholinguistics

Tutorials

Garance PARIS

Winter Semester 2011/2012

Roadmap

♣ Theories of sentence processing:

♣ modularity, parsing strategies, information sources, reanalysis

♣ Symbolic parsing models:

♣ incremental parsing, ambiguity resolution, memory load, probabilistic models

♣ Probabilistic parsing models:

♣ Symbolic parsers augmented with probabilities, derived from experience

♣ Connectionist models

♣ Distributed models of language learning and language processing

Matthew W. Crocker

Part I

Part II

Part I: Symbolic Models of Parsing

- Implement simple models:
  Bottom-up (shift-reduce), top-down,
  left-corner arc-standard and arc-eager

- Assess them using a simple *linking hypothesis:*
  Memory load

- Theory: Lecture 2

Part II: Connectionist Models

More information later

Modeling

- Forces clear specification of theories instead of vague descriptions by requiring sufficient detail for implementation

- Tests the coherence of a theory, especially when interactions between its parts becomes complex

- Generates testable predictions

- Permits manipulations that may not be possible in an experiment (lesioning, long term changes, e. g. learning vocabulary in a foreign language)

COGENT:
A Graphical Environment for Cognitive Modeling

- Idea: Simulate experiments (e. g. self-paced reading) and try to link the results to human data
- Box and arrow diagrams
  + a Prolog-like programming language
- Guided tour at
  http://cogent.psyc.bbk.ac.uk/tour/
- Runs on Mac, Windows and Linux (?)
- Intended for researchers, not end-users (devel-ware)
  - NOT very stable!
  - You should save your work often and frequently create new models and/or archives

COGENT in the Lab

- Boot/restart computer, turn on screen
- In the Grub menu containing operating systems, select MS Windows (be quick!)
- Login as `cogent`, making sure you select "`COLI-CIP`"
- Wait for startup script to finish executing
    - A fresh copy of `C:\LOCAL\cogent` is copied from the server

    - User data will be **DELETED** when you log out!

Saving Data

Three possible ways:

- Connect to your Linux home directory:
  - Open the Windows Explorer (not Internet Explorer)
  - In Address bar, enter \\smb
  - At the prompt, enter user and password to connect to your Coli account
  - Create your own Cogent directory wherever you like in your home directory
  - At the beginning/end of tutorial, copy files back and forth between your home directory and the local COGENT directory

Saving Data (continued)

Other possibilities:

- Insert a USB stick
- Use SFTP to connect to another machine
  - Main servers on which you can access your Coli account: `login` and `login2`
  - From the computer lab, only login is accessible

Use of Other Computers

- You can also use COGENT on your laptop in class if you wish
- COGENT runs on `login` (Linux) and can be accessed via SSH and X forwarding, but this can break any day due to Ubuntu upgrade (also, please keep an eye on the server load!)

Prolog Syntax: Term Types

- Atoms: Begin with a lower-case letter
  e. g. `socrates`
- Variables:
  - Begin with a capital or "_" (underscore)
    e. g. `Word, _G9876`
  - Anonymous variable: "_" (underscore)
    (A variable which you do not care to give a name to,
    often because it does not occur more than once)
- Complex terms: e. g. `love(john,mary)`
  - functor: `love`
  - arguments: `john, mary`
  - arity: number of arguments (`love/2`)
  - nesting:
    ```
    word(noun(sg(dog),pl(dogs)))
    word(noun(dog,sg))
    word(noun(dogs,pl))
    ```

Prolog Syntax: Lists (1)

- Basic notation: [element1, element2, element3]
- Used to represent sentences:
  [the, cat, chases, the, dog]
- Non-empty lists always consist of the list's first element
  and its tail, which is in itself a list:
  [element1 | [element2, element3]]

```
        [john, loves, mary]


   john       [loves, mary]


           loves     [mary]


                mary        []
```
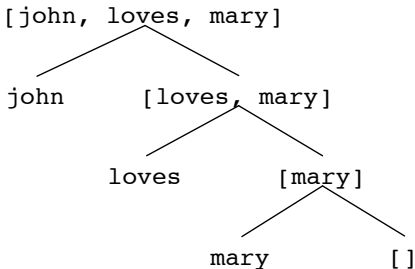
Prolog Syntax: Lists (2)

- The inbuilt operator "|" separates
  the first element from the rest:
  [Head|Tail] or [First|Rest]

- The last element is special: The empty list ("[]")
    - It cannot be taken apart and does not unify with
      anything else
    - A singleton (one-element list) like [element3] is
      therefore really equal to [element3 | []]
    - When we process a list element-wise and reach [],
      we know we have reached the end of the list
      (we're done)

Prolog Syntax: Knowledge

- Facts (statements which are true):
  - Example: `human(socrates).`
  - Used for words in the lexicon: `n(man). adj(tall).`
  - Note the dot at the end

- Rules:
  - Example: `mortal(X) :- human(X).`
    "All humans are mortal"
    (if X is a human, then X is mortal)
  - ":-" denotes implication:
    If the right-hand-side is true,
    then the left-hand-side is also true.

Prolog Syntax: Unification

- Equating two terms which have the same structure, thereby setting the variables in one term to the corresponding values in the other term
- As a consequence all other occurences of a given variable are also bound with the same value
- Example:
  Given `human(socrates).`

  `mortal(X) :- human(X).`

  `human(X)` can be unified with `human(socrates)`
  and therefore we learn that `mortal(socrates)`
- Often used to bind variables

Organizational

- Tutorials: Wednesday 14–16h
- Presence in the tutorial session is obligatory unless excused (e-mail)
- Part I has approximately 6 tutorials + mini-projects (2 sessions)
- Students are expected to finish each worksheet by the next session (4–5 hours of homework a week until Christmas)

Organizational, continued

- Timely completion of all tutorials and mini-project is required to sit the exam
- For each tutorial, submit an archive of your model and/or report by Monday
- Tutorials will be marked on a pass or fail basis (no grades)
- Questions/problems: Please contact me early on to arrange a meeting if necessary

Coli Accounts

- Department policy:
  Your Coli e-mail is your official contact address

  - Even if we work under Windows most of the time,
    you need a Coli account
  - Werner Saurer sends potentially important
    information about exams to `students@coli`
  - You must read your Coli e-mail!
    (especially external students)

- Assignments must be submitted using your Coli
  address

  (But you can forward your e-mail to another address
  + set up other software (Gmail, …) to display your

  Coli address as a sender)

Coli Accounts

Account configuration:

- See the System administrator's wiki:
  `http://www.coli.uni-saarland.de/sg/`
- Changing your password:
  `http://wiki.coli.uni-saarland.de/wiki/index.php/Password`
- Forwarding E-Mails:
  `http://wiki.coli.uni-saarland.de/mail2/index.php/Main_Page`
- Help: Ask a friend, the Fachschaft or me

Course Mailing List

We have a mailing list for the course:
`comppsych@coli...`

Frequently Asked Questions

`http://www.coli.uni-saarland.de/~gparis/Cours/`
`CP_Tutorials/faq.php`
Including: How to create archives in COGENT
(for submitting your tutorials)