Knowledge Engineering for NLP

January 14, 2008

Test suites

Overview

- General announcements
- Evaluation in computational linguistics
- Uses of test suites
- Evaluating precision grammars
- Levels of adequacy
- Test suite software ([incr tsdb()] demo)
- Time for questions about lab 2

Ask more questions!!

- About: lab instructions, phenomena in your language, the Matrix, the LKB, HPSG
- Who’s using the FAQ?
- Who spent $>$ 10 minutes figuring out an LKB error?

Questions from Lab 1 (1/5)

- How to put in an identity constraint and a specific value?
- Why is `COMPS & [ FIRST #first ]` ill-formed?
- How could we do NP$_i$?

Questions from Lab 1 (2/5)

- What happens to the types and rules you type in?
- What are edges and what are they good for?
- What is the parsing algorithm the LKB is using? Why is it written in LISP?
- Design decision or design limitation to disallow unbounded lists?
- Technical limitations (e.g., non-ascii) when doing new grammars?

Questions from Lab 1 (3/5)

- Can the node labels “know” which node is topmost? If not, how else do you distinguish VP from S?
- Is it better to have one unified label for VP? How could you do that?
- Why would the ordering of the labels make a difference?

Questions from Lab 1 (4/5)

- How is it that *null* is understood without any further specification? And where is there a detailed description of the FIRST and REST features for lists?
- What's the difference between this list recurion and diff lists?
- Can we do optionality and Kleene stars in tdl?
- Can we access ranges of lists e.g., COMPS[0,5]?

Questions from Lab 1 (5/5)

- What are the relative merits between binary-branching and flat analyses? Are the binary branching trees theoretically correct? Why does (theoretical) HPSG in general do flat structures?
- Can we adapt feature structures and unification to something less absolute?
- Last quarter, we were pushing most of the detail into the lexicon. Are we still doing that?

Evaluation and computational linguistics

- Why is evaluation so prominent in computational linguistics?
- Why is it not so prominent in other subfields of linguistics?
- What about CS?

Uses of testsuites

- How far do I have left to go?
  - Internal metric
  - Objective comparison of different systems

    → How do you evaluate precision grammars?
- Where have I been?
  - Regression testing
  - Documentation

Kinds of test suites

- Hand constructed
  - Controlled vocabulary
  - Positive and negative examples
  - Controlled ambiguity
- Corpus based
  - More open vocabulary
  - Greater ambiguity
  - Haphazard ungrammatical examples
  - Application-focused
- Which kind for which use?

Evaluating precision grammars

- Coverage against a corpus
  - Which corpus?
  - Challenges of lexical acquisition
- Coverage of phenomena
  - How does one choose phenomena?
  - What did TSNLP do?
- Comparison across languages

Levels of adequacy (1/2)

- grammaticality
- “right” structure
- “right” dependencies
- exact match semantics

Levels of adequacy (2/2)

- Only legitimate parses (how do you tell?)
- Some set of parses including the preferred one
- Preferred parse only/first
- Preferred parse within first N

Test suite software

- What does the LKB batch parse utility do?
- What else would you like it to do?
- → [incr tsdb()] demo

Overview

- General announcements
- LKB Formalism con't
- Evaluation in computational linguistics
- Uses of test suites
- Evaluating precision grammars
- Levels of adequacy
- Test suite software ([incr tsdb()] demo)
- Time for questions about lab 2