Knowledge Engineering for NLP

January 5, 2009

Introduction, overview

LKB Formalism

Overview

- The BIG Picture
- The LinGO Grammar Matrix
- Other approaches
- Goals (of grammar engineering, this course)
- Course requirements/workflow
- Pick a language, any language
- Components
- LKB demo, LKB formalism

Course URL

`http://courses.washington.edu/ling567`

The BIG Picture: Precision Grammars

- relate surface strings to semantic representations
- distinguish grammatical from ungrammatical sentences
- knowledge engineering approach to parsing
- can be used for both parsing and generation

The BIG Picture: Applications

- language documentation/linguistic hypothesis testing
- MT
- automated email response
- augmentative and assistive communication
- computer assisted language learning
- human-machine collaboration
- IR (“semantic search”)
- …

The BIG Picture: Challenges

- efficient processing (Oepen et al 2002)
- ambiguity resolution (Toutanova et al 2005)
- domain portability
- lexical acquisition (Baldwin 2005)
- extragrammatical/ungrammatical input
- scaling to many languages

The BIG Picture: Hybrid approaches (1/2)

- Naturally occurring language is noisy
  - Typos
  - “mark-up”
  - Addresses & other non-linguistic strings
  - False starts
  - Hesitations
  - …
- Allowing for the noise within the grammar would reduce its precision
- And then there’s ambiguity, unknown words, ...

The BIG Picture: Hybrid approaches (2/2)

- Combine knowledge engineering and machine learning approaches:
  - Statistical parse selection
  - (Statistical) named entity recognition and POS tagging in a preprocessing step (for unknown word handling)
  - Tiered systems with a shallow parser as a fall back for the precision parser
- Coming the other direction, deep grammars can provide richer linguistic resources for training statistical systems (e.g., MT systems).

The LinGO Grammar Matrix (1/3)

- One of the primary impediments to deploying precision grammars is that they are expensive to build.
- The Grammar Matrix aims to address this by providing a starter-kit which allows for quick initial development while supporting long-term expansion.
- The Grammar Matrix also represents a set of hypotheses about cross-linguistic universals.
- More recently, the Grammar Matrix has added typologically-grounded “libraries” exploring the range of variation in certain phenomena.

The LinGO Grammar Matrix (2/3)

- A sampling of hypotheses:
  - Words and phrases combine to make larger phrases.
  - The semantics of a phrase is determined by the words in the phrase and how they are put together.
  - Some rules for phrases add semantics, and some don’t.
  - Most phrases have an identifiable head daughter.

The LinGO Grammar Matrix (3/3)

- More hypotheses:
  - Heads determine which types of arguments they require, and how they combine semantically with those arguments.
  - Modifiers determine which kinds of heads they modify, and how they combine semantically with those heads.
  - No lexical or syntactic rule can remove semantic information.

Other approaches

- The DELPH-IN consortium specializes in large HPSG grammars.
- Other broad-coverage precision grammars have been built in/by/with:
  - LFG (ParGram: Butt et al 1999)
  - F/XTAG (Doran et al 1994)
  - ALE/Controll (Götz & Meurers 1997)
  - SFG (Bateman 1997)
- Proprietary formalisms at Microsoft and Boeing.

Goals: of Grammar Engineering

- Build useful, usable resources
- Test linguistic hypotheses
- Represent grammaticality/minimize ambiguity
- Build modular systems: maintenance, reuse

Goals: of this course

- Mastery of tfs formalism
- Hands-on experience with grammar engineering
- A different perspective on natural language syntax
- Practice building (and debugging!) extensible system
- Contribute to on-going research on multilingual grammar engineering

Course requirements/workflow (1/2)

- Mondays lecture, Wednesdays discussion
- Office/lab hours on Fridays (typically)
- Weekly lab assignments, posted Monday evenings, due Fridays (via CollectIt)
- Be sure to start the lab before class on Wednesday, so you can bring useful questions.
- At least half of each lab grade will be on the documentation.
- No exams; front-loaded.
- “Uncheatable”

Course requirements/workflow (2/2)

- Week 1: Getting to know the LKB (English exercise); pick your language
- Week 2-4: Constructing your test suite, iteratively customize a starter grammar
- Weeks 5: Build out your grammar
- Week 10: MT extravaganza

Surviving this course

- Communication is key: Please ask questions!
- Use GoPost (link on course page)
- Read (and contribute to!) FAQs, glossary ($\rightarrow$ demo)
- EB's lab hours
- The 10 minute rule

Pick a language, any language

- Each student must pick a different language.
- Previous languages on the wiki, under LanguagesList.
- No English; non-Indo European preferred.
- Consider using an ascii transliteration
- Languages with complex morphophonology require abstraction (assume a morphophonological processor)
- Pick a language with a good descriptive grammar available.

Components

- HPSG: theoretical foundations
- LKB
- Grammar
- Emacs: editor, interaction with LKB
- [incr tsdb()]

Components: LKB

- tdl reader
- parser
- generator
- interactive unification
- grammar exploration tools

Components: Grammar

- A set of tdl files:
  - Grammar Matrix core
  - Additions from customization system
  - Your additions
- Actually separated into:
  - type defintions
  - instances of grammar rules & lexical rules; lexicon
  - root symbols; abbreviations
- Lisp code for LKB interaction

Components: [incr tsdb()]

- Pronounced “tee ess dee bee plus plus” (or “the fine system”)
- Loading in test suites
- Running test suites
- Comparing competence over time

Overview

- The BIG Picture
- The LinGO Grammar Matrix
- Other approaches
- Goals (of grammar engineering, this course)
- Course requirements/workflow
- Pick a language, any language
- Components
- LKB demo, LKB formalism