

Test suites, [incr tsdb()]

Ling 567

April 8, 2013

Overview

- Questions from Lab 1
- Evaluation and computational linguistics
- Evaluation and precision grammars
- Test suites and precision grammars
- Our test suites
- Features of `[incr tsdb()]`
- Look at Lab 2 instructions

Ask more questions!

- This class is not designed so that you can complete the work on your own with the information provided.
- I'm *relying* on you to ask questions, and not spend lots of time stuck. The 10 minute rule is for real!
- The questions you all turned in as part of Lab 1 were excellent --- but of course could have been much more timely on GoPost.

Questions from Lab 1: HPSG and analyses

- Why is there a distinction between subject and specifier? (Why have both SUBJ and SPR? What's SPEC?)
- What's the difference between LOCAL and the rest of the structure, and is there a difference between C-CONT and CONT?
- Some of the lexical rules (same-xx-rule) use DTR as a feature, but I couldn't find a definition for it anywhere. Where does it come from?
- What's KEYREL for?
- What's XARG about? What about HOOK?
- Will we need to investigate the ins and outs of features that we didn't study in 566 ourselves, or will we get to them during this class?

Questions from Lab 1: Grammar Matrix

- It seems like this grammar is massive for such a small lexicon. Will the grammars grow exponentially or linearly?
- The grammar files seem really complex to me. How do you write the grammar so it could be easier to read and maintain later on?
- How would I fix the grammar to parse *Dogs danced.*?
- What is "binary-phrase"?

Questions from Lab 1: Grammar Matrix

- How much of this grammar are we expected to follow when building our own?
- The textbook has a different grammar, and Copestake indicates that features and values may vary quite a bit. However, I assume that our grammars should fit into the Grammar Matrix, and thus must follow the LinGO template to some degree. I'd just like to get a better idea of what resources we can assume are available to us and what we'll get to (or have to) develop on our own.

Questions from Lab 1: Grammar Matrix

- It's a bit confusing that the rules `singular-lex' and `plural-suffix' are a noun lexical rule while `pl-lex' is a verb lexical rule. Is it the matrix's default to not make a clear naming distinction between noun and verb lexical rules?

Questions from Lab 1: Misc

- I'm having a hard time determining how these feature structures are created.
- Can I search for sentences with the same semantics as a parsed sentence? (Right click graph, hit "Generate".)
- Further explanation of $\langle ! ! \rangle$ and how it interacts in the type hierarchy would be nice.

Questions from Lab 1: LKB software

- I am curious to know how LKB processes a grammar. If there is a step by step mode, that would be really fun!
- I am wondering if there is a way to get more information on how rules applied and inherited without guessing which ancestor rule a particular constraint is based on.
- Is there a way to ask LKB to explicitly show which constraints were used in the construction of the feature structure/parse tree?

Questions from Lab 1: LKB software

- I think it would be helpful if you could open up the AVM for two different strings, and have the differences between the two highlighted. Is there a way to do something like this?
- Why don't we have a way for the computer to tell us the chain of identities directly?

Questions from Lab 1: Grammar engineering

- One more: are grammars always this much of a treasure-hunt or are the type hierarchies purposefully obfuscated and complicated for this exercise?

Overview

- Questions from Lab 1
- Evaluation and computational linguistics
- Evaluation and precision grammars
- Test suites and precision grammars
- Our test suites
- Features of `[incr tsdb()]`
- Look at Lab 2 instructions

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?

Intrinsic v. extrinsic evaluation

- Intrinsic: How well does this system perform its own task, including generalizing to new data?
- Extrinsic: To what extent does this system contribute to the solution of some problem?
- Examples of intrinsic and extrinsic evaluation of parsers?

Test data

- Test suites
 - Hand constructed examples
 - Positive and negative examples
 - Controlled vocabulary
 - Controlled ambiguity
 - Careful grammatical coverage

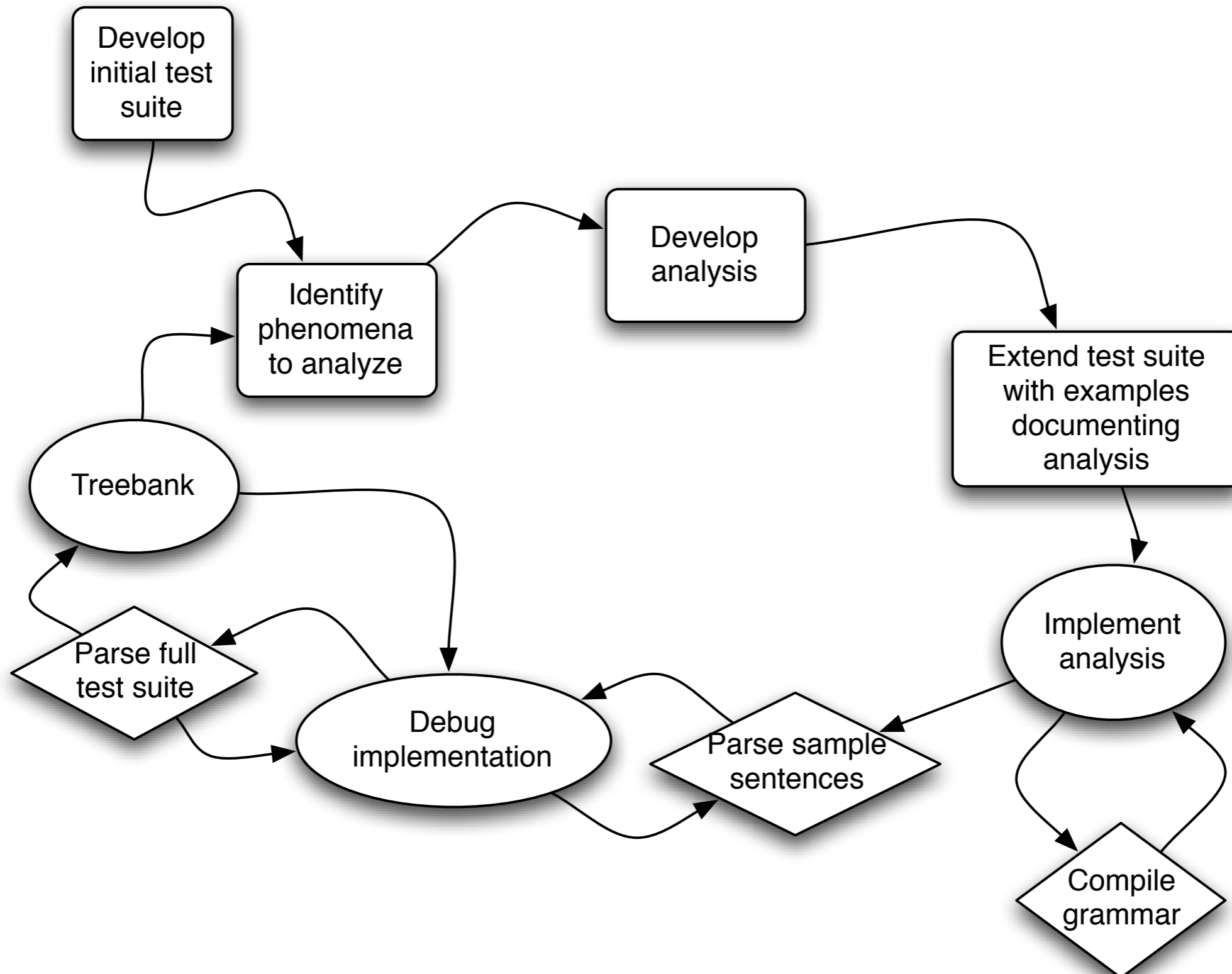
Test data

- Test corpora
 - Naturally occurring
 - More open vocabulary
 - Haphazard ungrammatical examples
 - Application-focused
- Which test data for which purposes?

Uses of test data

- How far do I have left to go?
 - Internal metric
 - Objective comparison of different systems
- Where have I been?
 - Regression testing
 - Documentation

Grammar engineering workflow



Evaluating precision grammars

- Coverage over some corpus
 - Which corpus?
 - Challenges of lexical acquisition
- Coverage of phenomena
 - How does one choose phenomena?
- Comparison across languages

Levels of adequacy

- grammaticality
- “right” structure
- “right” dependencies
- “right” full semantics
- only legit parses (how can you tell?)
- some set of parses including the preferred one
- preferred parse only/within first N

Our test suites

- Map out territory we hope to cover
- Include both positive and negative examples
- Serve as an exercise in understanding the description of the language
 - IGT format
 - Creating examples where necessary

[incr tsdb()] basics

- [incr tsdb()] stores test suite profiles as (plain text) relational databases: Each is a directory with a fixed set of files in it.
- Most files are empty.
- A profile that has not been processed has only two non-empty files: item (the items to be processed) and relations (always the same)
- Once the profile has been processed, the result of the processing is stored in some of the other files (in particular, parse and result)

[incr tsdb()] basics

- A test suite *skeleton* consists of just the item and relations files and can be used to create new test suite profiles
- [incr tsdb()] allows the user to compare two profiles to see how they differ
- It can also produce graphs plotting summary data from many profiles to visualize grammar evolution over time
- -> Demo

Overview

- Questions from Lab 1
- Evaluation and computational linguistics
- Evaluation and precision grammars
- Test suites and precision grammars
- Our test suites
- Features of `[incr tsdb()]`
- Look at Lab 2 instructions