

Test suites, [incr tsdb()]

Ling 567

January 11, 2010

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 are the semantic and syntactic components combined?
- Where do most of these new features originate?
- I'm curious about the placement of the features in the grammar and exactly what some of the features do. For example, I'm familiar with PER, but what exactly are XARG and PNG? Why is PER in XARG? I'm sure we will study this in more detail as the class progresses.

Questions from Lab 1: Type hierarchy

- What is the point of having some of the definitions being only 1 other supertype? This seems to just change the name of the type.
- What is the purpose of the basic types, such as `avm` and the many different lists?
- There are redundant supertypes declared in some of the inheritance hierarchies. Is this an artifact of incremental design or is it by design?

Questions from Lab 1: Features & misc

- Why are HEAD-DTR and NON-HEAD-DTR structures explicit in the grammar? Doesn't ARG-ST or VAL features provide paths to daughter structures?
- What is the difference between SUBJ and SPR, and why do we need both?
- What does it mean when there are a list of features separated by periods? For example “NON-HEAD-DTR.SYNSEM.LOCAL.CAT.MC”
- Is the LKB case sensitive? What happens to punctuation in input strings?

Questions from Lab 1: LKB software

- In cases where I don't know where to look for the information, how could I figure out that I should look at a lexical entry rather than a rule?
- Will I possibly have time this quarter to write a script that would simultaneously display all of the constraints on a given type so I can thoroughly investigate a type more efficiently? Corollary -- Is such a display feasible in text format?
- What's the difference between full and local avms?

Questions from Lab 1: Grammar engineering

- How can one derive the grammar from texts and similar related languages? How would modeling this language aid the researches trying to parse ancient/crumbling texts?
- The matrix.tdl definitions look like they can handle subordinate clauses. Does the customization system not handle these because the matrix grammar does not handle them properly, or because there is not a good way to write a customization page for them?

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

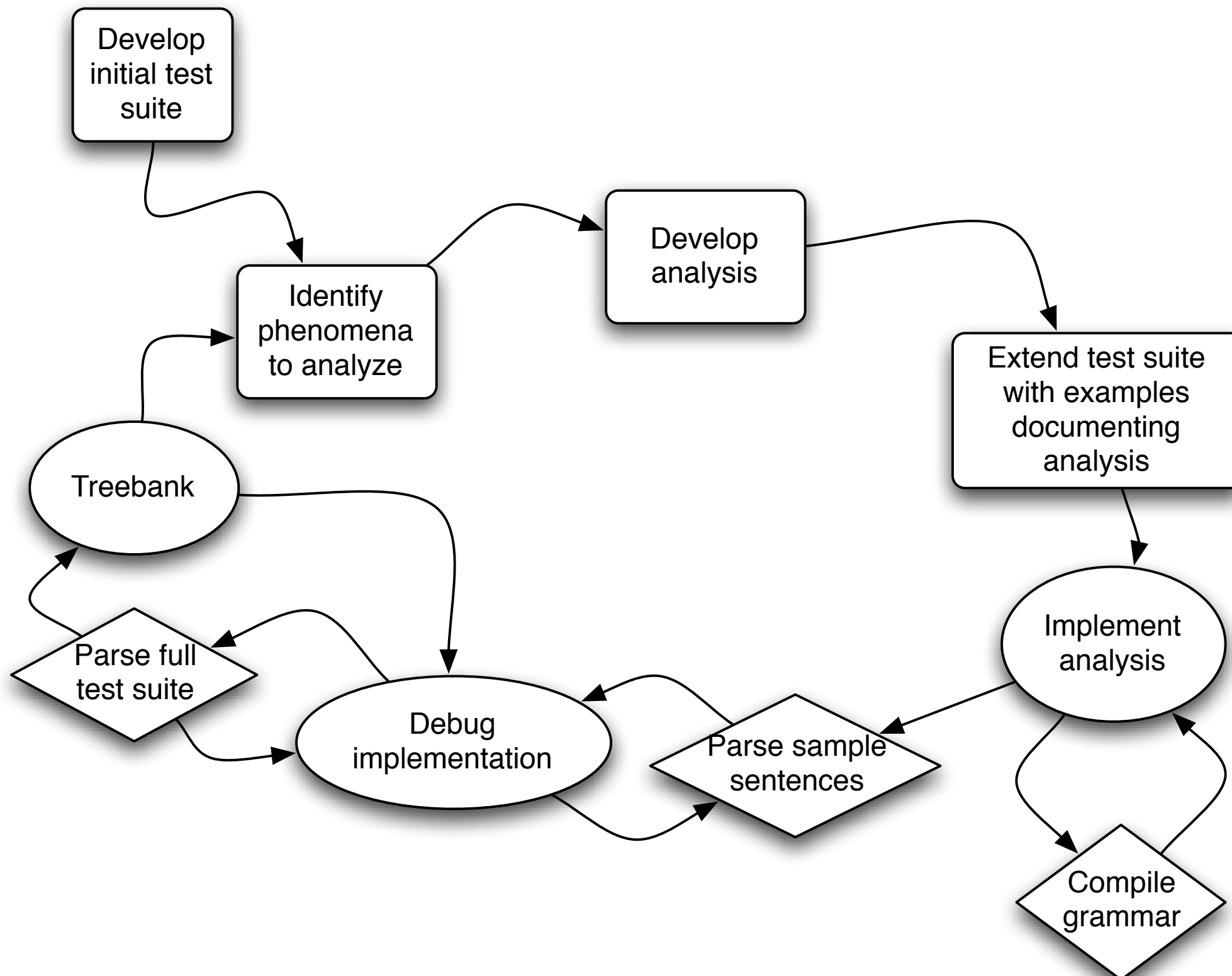
Test data

- 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