Final project ideas Computational Linguistics (LING 455)

This document has some suggestions to get you started thinking about your final project. To re-emphasize something I mentioned in Friday's class (Oct 29), the project is not intended to be huge — roughly on the order of two homework assignments (albeit ones you pick and design yourself) is a reasonable goal to keep in mind.

Your topic does not have to come from this list; you have a fair amount of freedom in what you choose to work on. Some of these topics are highly general. Your proposal (250–500 words) should be concrete and specific about the direction you have chosen and what led you to it, what you hope to accomplish, and how.

1 Programming project ideas

I will be happy to consider any program that uses Haskell to handle linguistic data (either analysis or production thereof), though my general expectation is that your project will build off some code or technique that we have discussed in class.

Your code should be accompanied by a brief writeup in plain text or pdf format which explains what your code does, and shows how it handles a few representative use cases, with enough specificity that I can take the program for a spin myself. Say a little bit about what you learned in the course of completing the project, and assess how well your code accomplishes the goal you began with. Here are some ideas:

- **FSA/Regexp conversion.** We discussed informally how one could convert between FSAs and regexps, but never explored a computational implementation. There are many techniques for translating between the two representations. Implement one of them (your program only needs to translate in one direction, not back and forth).
- **Other techniques for Regexp parsing.** We implemented a `match` function for `Regexp`'s one way, but others approaches are possible, some of them much more efficient. Learn about another way to parse regexps, and implement it.
- **Strictly Local Grammars.** Regexps and FSAs are equally powerful, in that they can characterize exactly the same languages/string patterns. Context Free Grammars (the next topic in our course) are strictly more powerful. *Strictly Local Grammars* (SLGs) are strictly *less* powerful. Learn about the use of SLGs in the analysis of linguistic data, and construct an implementation of some SLG-based analysis. **(This would also make an excellent paper topic.)**
- **Finite state methods in semantics.** Semantic automata have been used to investigate natural language quantifiers. This has yielded productive insights into the computational complexity of different kinds of quantifiers, and such results have been related to psycholinguistic measures. Learn about and implement an automata-theoretic approach to the semantics of quantification. **(This would also make an excellent paper topic.)**
- **Other methods for parsing CFGs.** We will discuss a chart-based approach to CFG parsing known as the CYK algorithm. Investigate and implement some other approach to CFG parsing (e.g., Earley parsing).
- **Phonological rules and FSTs.** As we discussed a bit in Week 9 (and will discuss a bit more in Week 10), phonological rules of the form 'A → B / C __ D' can be implemented as FSTs. Write a program that can covert any phonological rule into a finite-state transducer. Implement the *composition* of two phonological transducers, an operation that applies one to an input string, and then applies another to the string output by the first. Consider the results this yields for cases in which *rule ordering matters*.
- **Other grammatical formalisms.** There are many syntactic frameworks that we cannot discuss in this course for reasons of time — Combinatory Categorial Grammar, Tree Adjoining Grammars, Linear

Indexed Grammars, Tree grammars and tree transducers, Type-logical Grammar, Minimalist Grammars, to name just a few. Learn about one of these frameworks, and implement a small parser in Haskell. **(This could also make an excellent paper topic.)**

- **Computational linguistic analysis.** A computational implementation of a theoretical linguistic analysis that you have learned about in one of your other courses. You might construct a finite-state implementation of some aspect of the phonology or morphology of some language, for example — or extend one of the techniques for parsing we examined in order to handle some new type of construction, such as *wh*-movement.

2 Paper project ideas

As mentioned above, several of the programming topics are natural candidates for papers, if the project is correspondingly more research-based and is not accompanied by an implementation:

- **Strictly Local Grammars.**
- **Finite state methods in semantics.**
- **Other grammatical formalisms.**

Here are some other potential topics:

- **Computational models of human parsers.** The computational implementation of parsers makes it possible to precisely characterize the complexity and/or difficulty of parsing certain constructions, and to potentially relate this to psycholinguistic measures (as observed, e.g., in eye-tracking, or self-paced reading studies of garden-path sentences). Research and write about some aspect of this domain of research.
- **Gradient models of grammatical competence.** The grammatical formalisms we have considered in this course are, by and large, ones that model grammaticality as all-or-nothing. It is increasingly clear, though, that non-categorical or *probabilistic* methods are appropriate to the study of linguistics, and not just the province of atheoretical applied work. It is likewise increasingly clear that such methods are not in substantial tension with the aims, priors, and techniques of mainstream theoretical work. Research and write about some aspect of gradient or probabilistic linguistics — some examples are probabilistic approaches to syntax (see Christopher Manning's "Probabilistic Syntax"), or stochastic approaches to phonology such as Stochatic OT, Harmonic Grammar, or weighted finite-state machines.
- **Computational analysis of some linguistic phenomenon.** Learn and write about a computational approach to some linguistic phenomenon that interests you. This is an open-ended topic, and it is important that your proposal is specific about what you plan to investigate, and includes some key works on your proposed topic that you have read, or plan to.
- **Learning.** Computational approaches to language have had a large influence on the study of how linguistic generalizations can be *learned*. A great deal of research investigates how learning is possible, what kinds of patterns are learnable, and what kinds of prior knowledge are required in order for successful learning to transpire. Especially closely related to the themes of this course (and quite accessible!) is Gildea & Jurafsky's work on automatic induction of FST's for phonological rules.
- **What are hidden Markov models?** How are they applied to a particular problem within computational linguistics?