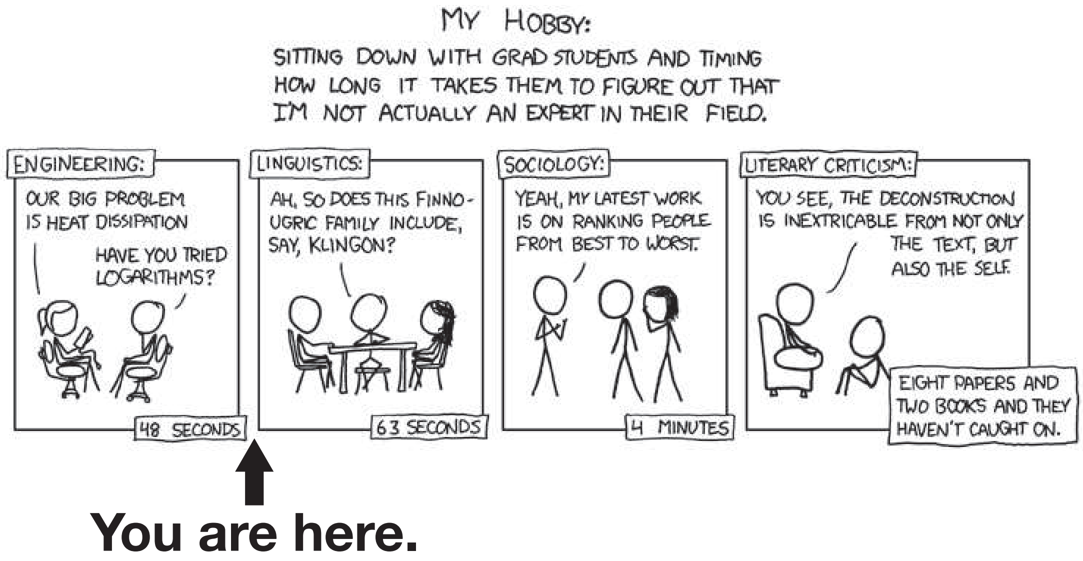
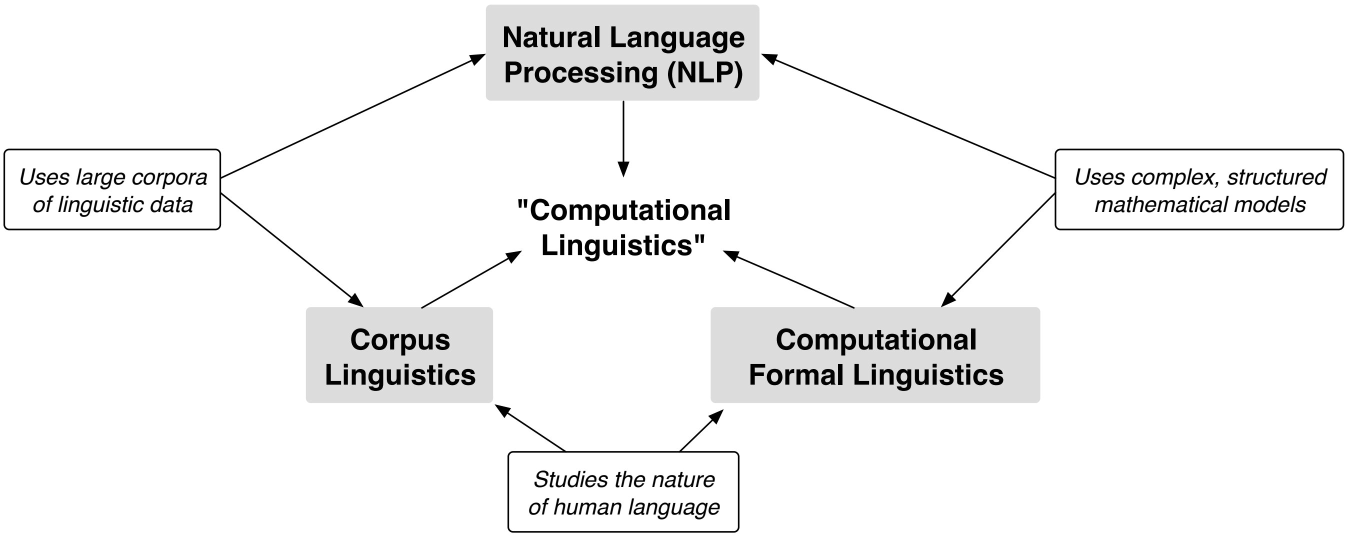
Computational Linguistics, Corpora, and NLP

Sam Bowman | Ling 1 | March 5, 2013


MY HOBBY:
SITTING DOWN WITH GRAD STUDENTS AND TIMING HOW LONG IT TAKES THEM TO FIGURE OUT THAT I'M NOT ACTUALLY AN EXPERT IN THEIR FIELD.
ENGINEERING:
OUR BIG PROBLEM IS HEAT DISSIPATION
HAVE YOU TRIED LOGARITHMS?
48 SECONDS
LINGUISTICS:
AH, SO DOES THIS FINNO-UGRIC FAMILY INCLUDE, SAY, KLINGON?
63 SECONDS
SOCIOLOGY:
YEAH, MY LATEST WORK IS ON RANKING PEOPLE FROM BEST TO WORST.
4 MINUTES
LITERARY CRITICISM:
YOU SEE, THE DECONSTRUCTION IS INEXTRICABLE FROM NOT ONLY THE TEXT, BUT ALSO THE SELF.
EIGHT PAPERS AND TWO BOOKS AND THEY HAVEN'T CAUGHT ON.
You are here.

1 What do you mean by Computational Linguistics?


Natural Language Processing (NLP)
Uses large corpora of linguistic data
Uses complex, structured mathematical models
"Computational Linguistics"
Corpus Linguistics
Computational Formal Linguistics
Studies the nature of human language

There is no fundamental reason why these three subfields need to be separate, but for practical reasons they are usually studied independently.

2 Natural Language Processing

(1) NLP is a subfield of artificial intelligence (a.k.a. applied machine learning); most research in Computer Science departments and in the research staffs of companies like Google, Microsoft.

(2) What are three specific problems that people in NLP work on solving?

3 Corpus Linguistics

(3) Corpus linguistics uses observations from large collections of data, called *corpora* (singular: *corpus*), to answer questions about human language.

(4) A corpus is *any* stored human language data, with or without linguist-added information.

(5) Is it a corpus:

  a. The complete archive of the New York Times?
  b. This handout?
  c. Ten audio interviews with strangers at the Caltrain station, with transcripts?
  d. All of Battlestar Galactica on DVD?
  e. This lecture?

(6) Corpus linguistics typically refers to descriptive work, done alongside theoretical work in one of the more traditional subfields.

(7) What other kinds of data (er... corpora) are out there might you want to draw on as a linguist?

(8) What makes a good corpus?

(9) Common types of corpus:

  a. Text...
    i. ...with syntactic parses (tree structures)
    ii. ...with semantic parses (logical representations of meaning)
    iii. ...with coreference information ("Have you seen Bob$_1$? I need to ask him$_1$ about something.")
    iv. ...with sentence-by-sentence translations
  b. Speech...
    i. ...with speaker background information
    ii. ...with transcripts ("so then I said...")
    iii. ...with narrow(ish) phonetic transcriptions ("sow ðɛn aj sɛd...")
  c. There are lots of big, well documented corpora for major languages (English, French, Arabic, Mandarin...), but it's possible to find some amount of data for nearly any written language!

(10) Some widely used corpora:

  a. Brown Corpus (1964): One million words of written American English, sampled from a broad range of genres.

b. Switchboard: Recorded phone calls between strangers in American English, with detailed transcriptions.

c. The Penn Treebank: Text from both of the above, with syntactic parses.

d. The Google Web Treebank: Text drawn from all over the modern English web, with syntactic parses.

e. CHILDES: Conversations between parents and young children of various ages.

f. Santa Barbara: Recordings of face-to-face interactions between Americans from a wide and balanced range of regional origins, ages, occupations, genders, and ethnic and social backgrounds.

(11) Using corpus data is a great way to start investigating almost any linguistic phenomenon, but it is easy to get misleading impressions.

a. If your corpus isn't drawn from the same genre or style of language you want to study...

b. If your corpus is drawn from written data, and you want to study spontaneous speech...

c. If you don't have a thorough understanding of the statistical tools you use...

3.1 A case study from Stanford: Prosody and porn stars

(12) From Stephanie Shih and Tyler Schnoebelen (Shih, 2012, ...):

a. Does phonology influence parents' choice of baby names?

b. Faceoook Names Corpus: Hundreds of millions of first–last name pairs.

c. First pass: Names alliterate (i.e. Peter Potts > Rodger Potts) more than would happen by chance.

d. Names avoid adjacent stresses (i.e. Súsan Smíth > Suzánne Smíth) more than would happen by chance.

e. (What would one need to do to prove this?)

f. If people were able to choose full names, without the constraint of fixed surnames, we would expect these effects to be stronger.
⇝ Study porn star stage names. (Work in progress.)

4 Computational Formal Linguistics

(13) Develop theories of the language faculty in the mind by building them as software.

What I cannot create, I do not understand. (Feynman, 1988)

(14) Why would we want to model linguistic theories computationally?

(15) Some common cases:

a. Natural logic proof systems that make logical deductions from sentences in natural language.

b. Simulated models of neural activation to model word recognition in low-level speech perception.

c. Chomskyan syntactic grammars of selected chunks of real languages.

(16) **Caveat!** Computational in this context refers to an *algorithmic perspective* on modeling, mare than it refers to the actual building of software models. Computational formal linguists will sometimes embark on large research projects that don't involve writing a single line of real computer code.

5 A case study: Optimality Theory

5.1 Optimality Theory in ten(ish) minutes

(17) Optimality Theory (Prince and Smolensky, 1993/2004): Replaced rules (Chomsky and Halle, 1968) in the mid-1990s as the dominant way of describing phonology.

(18) The basic idea:

a. The grammar expresses what it wants outputs to look like, rather than explicitly listing the steps used to produce those outputs.

b. An optimization function takes those preferences and finds *the best possible way* to satisfy them.

c. It's usually not possible to satisfy every preference, so the language needs to specify which ones are most important to satisfy. In most versions of Optimality Theory, the preferences—called constraints—are ranked.

(19)

Reading linguistics in dorm room	Get sunshine	Avoid 'that guy'	Avoid mud on jeans	Avoid walking too far
a. Stay where you are	*			
b. Go read at the dorm lounge		*		*
c. ☞ Go read on the lawn			*	*
d. Go read at Green Library	*			***

(20)

/dɔg+s/	*[+Voi][-Voi]	Don't Delete	Don't Insert	Preserve[Voi]
a. [dɔgs]	*			
b. ☞ [dɔgz]				*
c. [dɔgɪz]			*	*
d. [dɔgɪs]			*	
e. [dɔg]		*		
f. [] (silence)		****		

(21) What output wins with this ranking?

/dɔg+s/	Preserve[Voi]	*[+Voi][-Voi]	Don't Delete	Don't Insert
a. [dɔgs]		*		
b. [dɔgz]	*			
c. [dɔgɪz]	*			*
d. [dɔgɪs]				*
e. [dɔg]			*	
f. [] (silence)			****	

(22) There is something special about *c* and *f*? (Think about what happens if you keep rearranging the rankings of the four constraints, and what you need to do to make one of them win.) These constraints are *harmonically bounded.*

(23) What's so great about Optimality Theory?

a. By far the most widely used approach to phonology.

b. Matches intuitions fairly well. It's often easier to come to a consensus about what kinds of words a language allows than about what kinds of processes the language uses.

c. Optimality Theory can predict typologies. Since constraints are assumed to be shared across languages, a claim about one language is a claim about all languages. (These predictions often work out disturbingly well.)

(24) But there are problems, right?

a. In theory, at least, you are drawing from an infinite pool of options. You could communicate the form /dog+s/ by saying [dogz], or you could communicate it by shouting the entire 1987 Iowa state tax code. Or you could communicate it by doing jumping jacks.

b. Finding the *best possible* output in some situation might not always be possible. You need to make sure that your grammar is set up in such a way that you can easily rule out infinitely large sets of candidates without having to consider them one-by-one.

c. Even when you know that it is possible to find an optimal output, it might be difficult to do that on paper, by hand.

(25) So. Computation?

a. Yes!

5.2 What does computational optimality theory look like?

The figures and representation used here are borrowed from Riggle (2009).

(26) Turn constraints into *regular expressions*: Patterns that a program can use to find out where the constraint is violated.

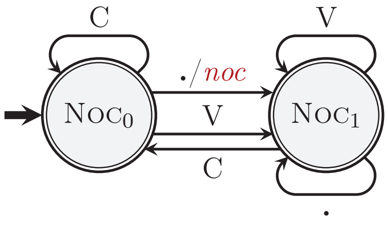
a. *[+Voi][-Voi] → `[+Voice][-Vocie]`

b. *Preserve[Voice] → `([+Voice]->[-Voice])|([-Voice]->[+Voice])`

(27) Or, for our present example of allowed and disallowed syllables we might use something even simpler, like this:

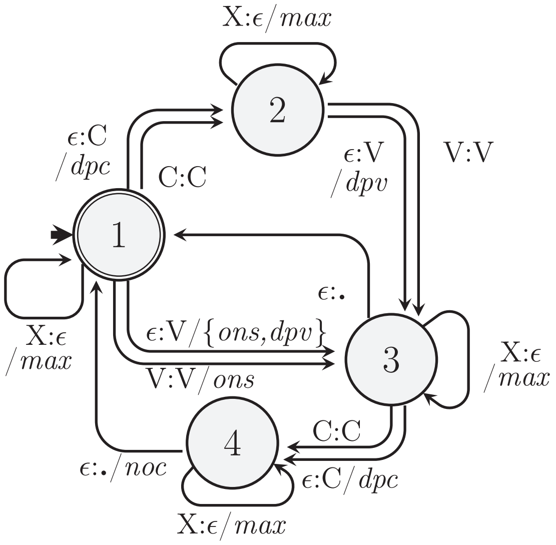
a. RequireOnset → `.V`

(28) This representation allows us to build constraints into graphs[1] like this:



(29) "." matches a syllable boundary, "V" matches any vowel, and "C" matches any consonant. "/*noc*" means that that the edge with that label assigns one violation of the constraint Noc. **What kind of phonological pattern does this constraint penalize?**

(30) From there, we can intersect these constraints into a single graph, which takes a candidate and calculates all of its violations of all the constraints:



(31) This is where this gets useful: It is possible to find all of the paths through the graph—all the candidates—that aren't harmonically bounded.

(32) There are only a small finite number of candidates like this.

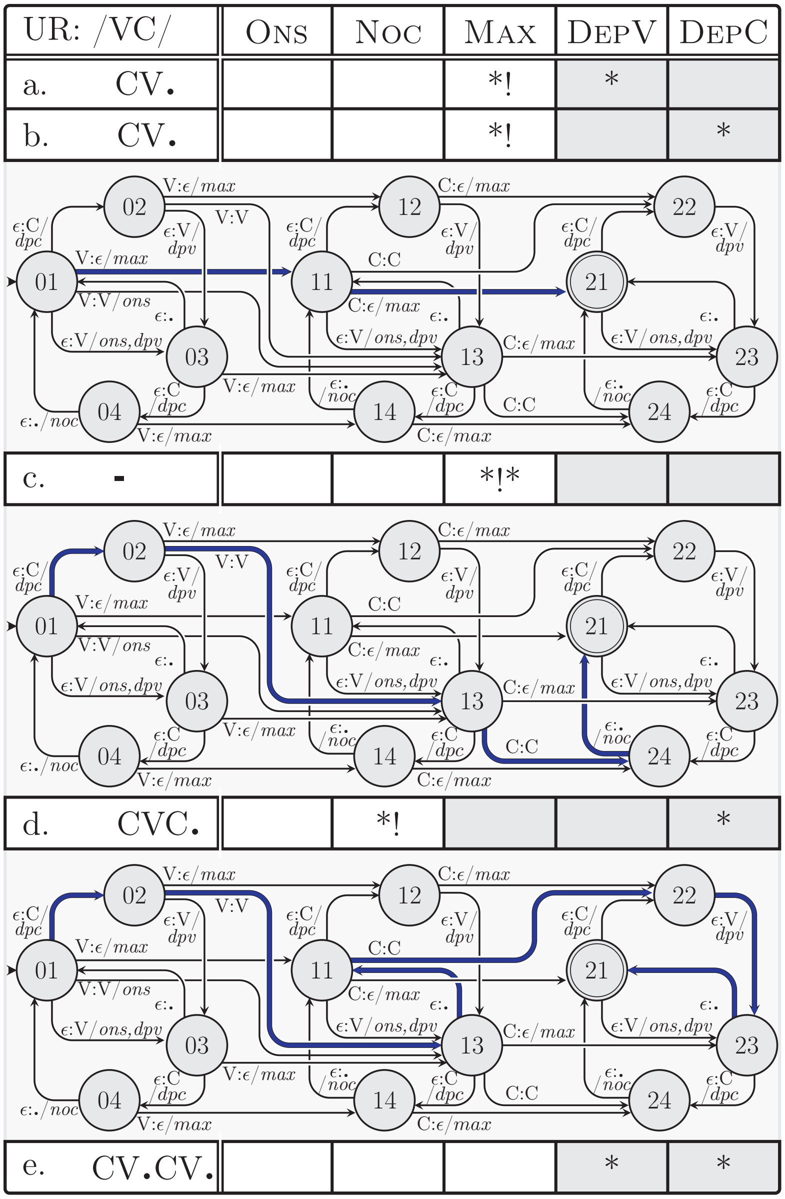
(33) This looks something like Figure 1.

(34) This lets you ultimately build typologies like this:

	/V/	/CCV/	/VC/	# rankings	%	/ɛlvɪs/?
Language type 1	[CV]	[CCV]	[CVC]	5	15%	[hɛl.vɪs]
Language type 2	[CV]	[CV.CV]	[CV.CV]	4	12%	[hɛ.lə.vɪ.sə]
Language type 3	[-]	[CV]	[CV]	2	6%	[vɪ]
Language type 4	[V]	[CV.CV]	[VC]	1	3%	[ɛl.vɪs]
...	...	...	...	...	...	...

(35) With this kind of information, you have a good sanity check on your theoretical claims:

[1] When I say graph, I'm using the word in the mathematics and computer science sense: A bunch of simple abstract objects with links (arrows) between them. This doesn't have too much in common with the *let's turn our data into a picture!* sense used in statistics and the experimental sciences.



UR: /VC/	Ons	Noc	Max	DepV	DepC
a. CV.			*!	*	
b. CV.			*!		*
c. -			*!*		
d. CVC.		*!			*
e. CV.CV.				*	*

Figure 1: All of the candidates that could be outputs for the input /VC/, given some ranking of these constraints. Bold arrows indicate which path through the graph each candidate represents.

a. Do the constraints and mechanisms that I propose get *every* output right for language x?

b. Do most of the language types that my proposals predict actually occur in the world?

c. Are the language types that I predict to be most common actually relatively common in the world?

(36) Want to play with these ideas with a real program? It's available for free as *PyPhon* (Bane et al., 2011) at:`http://code.google.com/p/clml` (There's not much of a user interface yet.)

6 Quick recap

(37) *Computational Linguistics* can refer to any of three different areas of research at the intersection between computational methods and linguistics:

(38) Doing natural language processing means using sophisticated machine learning techniques to enable computers to understand or produce language in useful ways.

(39) Doing corpus linguistics means answering questions about language by looking huge volumes of actual language, often with the help of statistical tests.

(40) Doing computational formal linguistics means taking your hypotheses about how humans do language, and testing them by seeing if you can make computers do language that way.

7 Why go into computational linguistics?

(41) Almost any aspect of linguistics can be studied computationally, and doing computational work allows you to be more confident of both your descriptive observations and your theoretical claims.

(42) Computational linguistics, especially NLP and computational formal linguistics, are central to both theoretical and applied artificial intelligence, and familiarity with computational linguistics is a great way to get any number of interesting and well-paid research jobs.

So: Stick around the department, and take a few math or computer science classes too!

References

Max Bane, Jason Riggle, and Samuel Bowman. PyPhon software package. `http://code.google.com/p/clml/`, 2011.

Noam Chomsky and Morris Halle. The sound pattern of english. 1968.

Alan Prince and Paul Smolensky. *Optimality Theory: Constraint Interaction in Generative Grammar.* MIT Press, 1993/2004.

Jason Riggle. Generating contenders. ms. University of Chicago. (ROA #1044), 2009.

Stephanie Shih. Linguistic determinants in english personal name choice. In *LSA 86th Annual Meeting*, 2012.

Comic: `xkcd.com`