Studies in the Linguistic Sciences
Volume 28, Number 2 (Fall 1998)

LINGUISTICS IN A COMPUTATIONAL WORLD

Daniel Jurafsky
University of Colorado, Boulder
jurafsky@colorado.edu

Linguists are talking about computers. What's up with that? What role do computers and computation play in linguistics in 1999? How are we currently using computational tools in linguistic research and in the linguistic curriculum? What about the job market for our students? I'd like to use this paper (a very faithful summary of a talk given at the Symposium *The Linguistic Sciences in a Changing Context* at the University of Illinois's Center for Advanced Study) to take a very brief glance at what's going on, computationally speaking, in the linguistic world. I'll begin with some high-level thoughts about the role of computation in linguistic research, past and present, turn to the job market, and then discuss computation in the classroom, both for general linguistics and for computational and corpus linguistics. I'll use examples from my experience at Boulder, and so the topics will be biased toward my own areas; the reader should of course fill in their own experiences. This paper is meant to start a discussion, not to provide a solution.

1. Computation in linguistic research: The computer as tool

The computer has been most obvious in linguistics in its role as a research tool. This is especially true in phonetics. Instrumental phonetics and laboratory phonology rely heavily on the computer for easy access to waveforms, spectrograms, spectra, and pitch traces, things which until recently had to be done on specialized equipment. The availability of digitized speech corpora has also played a role in laboratory phonology, making it easier to develop and apply theories like TOBI. Such signal analysis software packages are now widely used on PCs, Macs, and UNIX platforms, although no one software tool runs on all three platforms. The computer has also played an obvious role in corpus linguistics. One very successful example of this has been the CHILDES corpus established by Brian MacWhinney (MacWhinney 1995), which has been an essential resource for modern studies in language acquistion.

What future areas of linguistics could be revolutionized by the use of computer tools and corpora? One important role for corpora is in what might be called *interface studies*; research on the interface between linguistic levels. If a corpus is

annotated at multiple linguistic levels, it is easy to ask questions about how a given structure at one level maps to a structure at another level. To this end a lot of recent research in our lab at Boulder has relied on the annotated Switchboard corpus of conversational English telephone conversations (Godfrey et al. 1992). Switchboard is unique first in its breadth: 2400 conversations, 2.4 million words, 200 hours of speech, 500 different speakers. It is particularly interesting, however, in the depth of its annotations; just over half the corpus (1200 conversations) was annotated by the Linguistic Data Consortium and others for:

Sociolinguistic variables (age, sex, and dialect of each speaker)
Speech disfluencies and repairs (Coded by Meteer et al. 1995 using the coding scheme of Shriberg 1994)
Part of speech tags (using the Penn Treebank tagset - LDC)
Dialog acts (using 60 categories such as Question, Statement, Acknowledgement, Backchannel, etc., by Jurafsky et al. 1997)

In addition, selected portions of the Switchboard corpus were coded for more labor-intensive information:

3.5 hours were phonetically hand-transcribed by Steven Greenberg and his team at ICSI/Berkeley (Greenberg et al. 1996).
400 conversations were parsed as part of the Penn Treebank project (Marcus et al. 1993)

The result of this effort has been a number of papers from our lab studying interface effects. For example Gregory & Michaelis 1998 (following Birner & Ward 1998) used the parsed portion of Switchboard to study pragmatic use constraints on syntactic constructions. Because the corpus is parsed, they could automatically select all the instances of topicalization or of left-dislocation to examine for relevant syntactic or pragmatic properties. Jurafsky et al. 1998 and Bell et al. 1999 used the phonetically transcribed portion of Switchboard to study the causes of reduction/lenition in English function words. The rich annotations in the corpus enabled them to show that function words are longer and less reduced when they occur just before disfluencies, when they are less probabilistically predictable, when the speaker is female or elderly, or when they occur turn-initially or turn-finally. Jurafsky et al. 1998b used the dialog-act labels to study the lexical and syntactic properties that characterize specific dialog acts (e.g., the 'micro-grammar' of assessments and of reformulations).

2. Computation in linguistic research: The computer as metaphor.

Computation has also played a less-obvious role in linguistics: as a source of metaphors for processing. Two very salient examples are UNIFICATION and OPTIMALITY THEORY. Unification is the fundamental operation of many modern linguistic theories of syntax and grew out of the convergent ideas of a number of computer scientists and linguistics working in Palo Alto. Martin Kay was working

at Xerox PARC with Ron Kaplan, looking for a way to revise ATN grammars to make them reversible for machine translation. The problem was that the contents of an ATN register could be changed arbitrarily; these changes made reversibility impossible. For example, a parser might place a sentence-initial NP in the subject register, but then move it to the object register after encountering the verb 'be' and a passive participle. Kay began to move toward a view in which registers could not be overwritten, only extended. Essentially he was converging on the idea of logical variables, although without realizing it at the time. Meanwhile, Fernando Pereira and colleagues at SRI International were working on unification in the context of definite clause grammars, a field that arose in computer science out of logic programming. The result of these two computational efforts led to an information-combination operation and to a new way of implementing linguistic knowledge as a set of constraints (Kay 1979, inter alia).

Optimality theory describes the fundamental operation of a recent view of phonology and syntax, and arose from the collaboration of Paul Smolensky (at that time a cognitive scientist/computer scientist) and Alan Prince. Smolensky had been working in the connectionist paradigm, viewing connectionist networks as ways to optimize well-formedness constraints expressed by the network weights. He began looking for an area of cognition that relied on well-formedness; grammar was the obvious candidate. Smolensky and Geraldine Legendre first applied this metaphor in examining how well a linguistic input fit the constraints imposed by a grammar (Legendre, Miyata, Smolensky 1990). Meanwhile, in 1988, Smolensky and Prince appeared together on a panel on 'Connectionism and Psychological Explanation'. Their joint work combined harmony theory and phonology, and was originally called Harmony-theoretic Phonology. At this stage the theory still had numbers (weights) on the constraints. In April of 1991, they replaced the numbers with a ranking scheme. Just as with unification, a new metaphor for the representation of linguistic knowledge arose from the interaction of computation and linguistics

What might the future hold for new computational metaphors in linguistics? A natural candidate for borrowing from computation is learning theory. An important focus of computational models of learning (machine learning) is how to combine bottom-up experiences in the world with top-down learning biases. A simple instance of this process is the PARAMETER SETTING model of learning used in some theories of syntax. Here the learning bias is very strong, and the learner's experience in the world only contributes minimal new information. Outside of linguistics, by contrast, modern theories of learning are based on a weaker learning bias combined with distributional information from the world. Such distributional models have become common in psycholinguistics and computational linguistics, particularly in LEXICAL SEGMENTATION FROM SPEECH (Saffran 1996, Brent & Cartright 1996), GRAMMAR INDUCTION (Stolcke 1994, deMarcken 1997) LEXICAL SEMANTIC LEARNING (Landauer & Dumais 1997, Lund & Burgess 1995), and

PHONOLOGICAL RULE INDUCTION (Gildea & Jurafsky 1996). The learning biases in such systems are varied and come from many sources. Many rely on Minimum Description Length (Brent & Cartright 1996, deMarcken 1997, Stolcke 1994). Woodward & Markman 1991 propose specific word-learning biases. Gildea & Jurafsky 1995, 1996 use phonological Faithfulness. Regier 1997 used non-linguistic (visual) information to bootstrap the learning meanings of spatial prepositions.

Gildea and Jurafsky 1996, for example, studied the problem of phonological rule induction by training a standard automata-induction algorithm to induce the English flapping rule. The algorithm was presented with the surface form of 50,000 words containing a flap, together with the underlying form of each word. They found that the standard algorithm was completely unable to induce the contexts for flapping. They then augmented the learner with a FAITHFULNESS bias that preferred underlying forms to be faithful to surface forms, all things being equal. The addition of this bias enabled the algorithm to successfully induce the English flapping rule.

The Linguistic Job Market

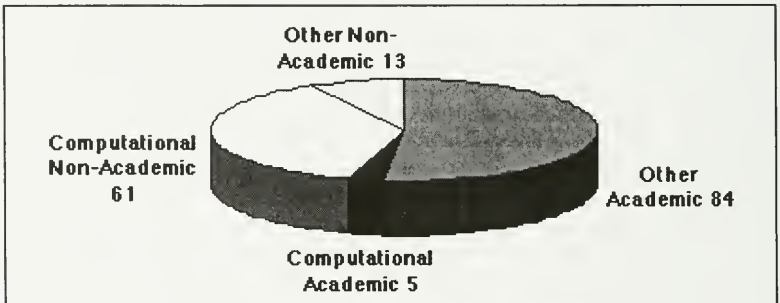


Figure 1: Computation in the Linguistic Job Market

Computation will clearly play an important role in linguistic research. But what about our responsibility to our students? What sort of computational jobs are available to our students, and how should we be preparing them? This section summarizes information on academic and non-academic jobs for linguistic graduates. First, I examined the on-line job listings from the LINGUIST LIST web site (http://www.linguistlist.org) and from LSA's Linguistic Enterprises web site (http://web.gc.cuny.edu/dept/lingu/enter.htm) in October 1998. I divided the jobs into academic/research (requiring Ph.D's; including tenure-track jobs, visiting lectureships, postdocs, and laboratory research jobs) and nonacademic, and into computational (requiring some computational experience more significant than the ability to use a spreadsheet) and non-computational. Figure 1 shows the result:

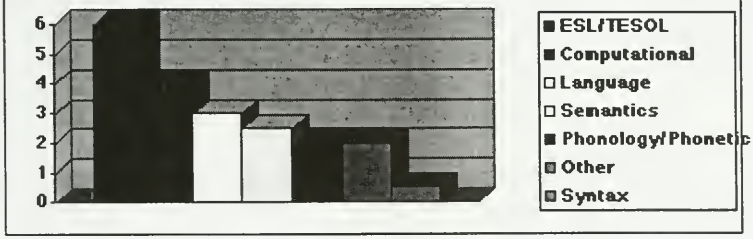
Over a third of the jobs advertised to linguists required computational skills. A typical Microsoft ad looked for:

> ...a linguist who can take DRT and semantic network outputs from an NL analysis system and provide the automatic inputs to a lower-level text realization component...
>
> Qualifications:
> the analysis of spoken interaction
> interface between linguistic and extralinguistic sources of information
> large-scale knowledge bases such as WordNet
> functional linguistics
> computational linguistics/NLP

A number of jobs looked for computational grammarians or dictionary-developers for foreign languages for speech recognition or grammar-checking.

What of the academic jobs? I looked at the job listings in the March 1999 LSA Bulletin. I coded only tenure-track faculty jobs in linguistics departments in the United States; there were a total of 20 jobs. Again, Figure 2 shows the prevalence of computational jobs.

Figure 2. Academic Jobs for Linguistics March 1999



This large percentage of computational jobs may be temporary, since I had assumed that few linguistics departments currently have computational linguists. In an attempt to check this assumption, I checked the number of departments with a faculty member who is cross-rostered in computer science. This algorithm will produce a conservative estimate, since some computational linguists may not be double-rostered. I found six departments: the Indiana University, the University of Colorado, the University of Delaware, the University of Maryland, the University of Pennsylvania, and the University of Southern California. Since there are also other schools with computational linguists (e.g., Ohio State, UCLA), the number of computational faculty is not insignificant.

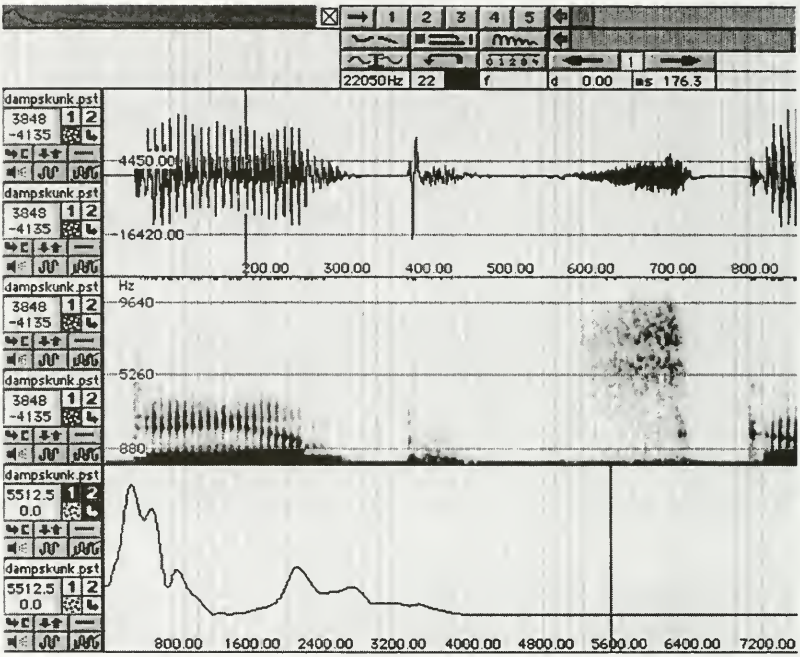


Figure 3. Eric Keller's Signalyze program for the Macintosh, showing a waveform, spectrogram, and spectrum (in the first vowel) of the author saying 'damp skunk' (the final 'unk' is cut off in this picture).

How should we be teaching computation?

If there are computational jobs, and if computation is important in linguistic research, how are we going to add computation to our curriculum? First, computation can be fruitfully used as a pedagogical tool in many linguistics courses that don't focus on computation. Second, we need to add specifically computational linguistics courses to our curriculum.

Let's begin with the first use of computation: as a pedagogical tool. Many linguistic courses have already been transformed through the use of corpora. Many phonetics courses, for example, including our course at Boulder, use Peter Ladefoged's online supplement to *A Course in Phonetics*, which includes Macintosh hypercard stacks for that book as well as the *Sounds of the World's Languages* stacks. This gives students a chance to play sounds from the IPA chart to help them learn them, to practice with performance exercises, and to hear rare phones in their lexical environments. Many phonetics classes also use signal

analysis software as a visualization aid in the acoustics component. We have been using Eric Keller's Signalyze software for lab homework assignments in our undergraduate phonetics class. Using ideas borrowed from John Ohala, we have them record their vowels and plot their own vowel chart, and also have them strip the *s* off of *skunk* to hear (and see) the devoiced *k*, as Figure 3 shows:

Many language acquisition classes, including ours, make use of CHILDES for student homework assignments and projects, as Figure 4 shows.

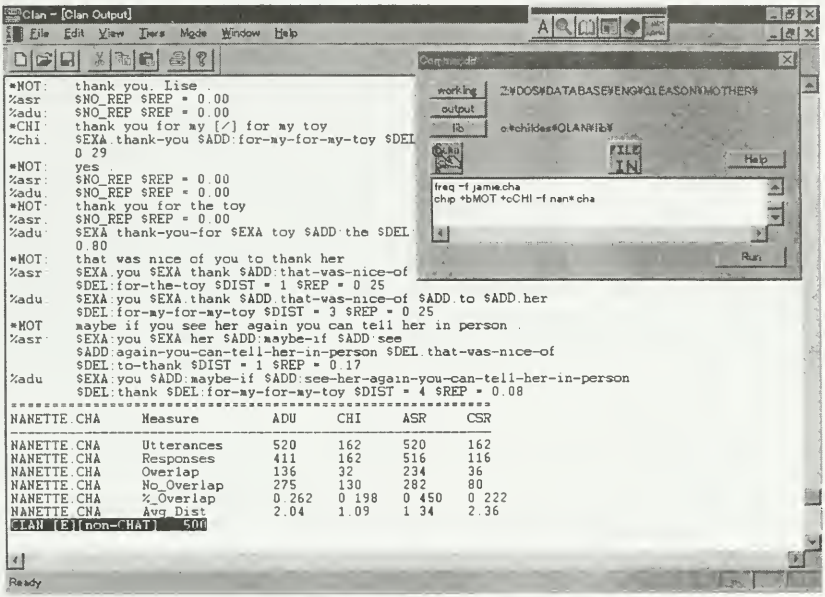


Figure 4. The CLAN program for searching the CHILDES database. This search shows the percentage of repetitions (self- and other-) in child and adult speech in a corpus.

Field methods classes regularly make use of software like SIL's LinguaLinks. We have recently been exploring ways to use web-based dialect vocabulary surveys in our intro sociolinguistics class, and ways to add homework assignments based on parsed corpora to our syntax classes.

Adding computers into the linguistic classroom is definitely not a time-saving device, and is not appropriate for every class. It is important when it can help make a difficult subject (like phonetics) easier to visualize, when it can help beginning students get access to rich data from corpora, and when it can help them exchange data in collaborative classes like Field Methods and sociolinguistics. Another important reason is that the linguistics classroom is an important

place to help overcome the gender gap in computer education, since we have such a good percentage of women students. The American Association of University Women Educational Foundation recently released a report on girls' education, 'Gender Gaps: Where Schools Still Fail Our Children'. They found that in 1996 girls made up only 17% of the high school students who took the Computer Science AP exam, and concluded that:

'While there are more girls taking computer classes, they tend to be in data entry, while boys are more likely to take advanced computer applications that can lead them to careers in technology'

Adding useful and challenging computer homework assignments into the general linguistics curriculum is a way to begin to overcome this gap.

This leads us to the second use of computers in the linguistics curriculum: in computational linguistics courses. Computational linguistics is such a new field that is not completely clear what it constitutes; different departments teach different things. Furthermore there is not yet a standard textbook (although I have high hopes for my about-to-appear textbook with James Martin (*Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*). Looking at the job market gives us some hint about what's needed:

speech recognition: especially phonetics, statistics, automata theory, programming
information retrieval (Web search engines) data-driven methods, programming
spell-checking and grammar checking – grammar-writing skills, programming
natural language and speech understanding – parsing, discourse and conversation, programming
machine-aided translation – syntax, semantics, use of on-line lexica and thesauri

While most of this is not covered in current courses, there is hope: of the top 20 linguistics departments in the 1995 NRC report, 70% have some sort of computational linguistics course. Most of these cover parsing (top-down, bottom-up, and chart), unification, finite-state automata, and semantic interpretation. The other 30% have a Natural Language Processing course in the Computer Science Department instead. At Boulder we have computational students take a course in Natural Language Processing in the Computer Science Department (taught by James Martin), but that course requires programming ability. This is often true of NLP courses, but should be thought of as a feature, not a flaw. The list above should make it clear that computational linguists must be able to program. Our solution is to use a *Computational Corpus Linguistics* course as the linguistics feeder course in which linguists learn to use corpora, and learn basic programming

techniques using Perl. They are then able to take NLP courses and more advanced corpus, computational, and speech processing classes. This also has the advantage that many students who don't choose computational linguistics as their main area still learn basic programming.

3. Conclusion

Computation will continue to play an important role in linguistics, as a source of new innovations in research metaphors, as a source of new data, as corpora and corpus tools open up new vistas on linguistic phenomena, and as a source of new pedagogical tools. Furthermore, since human-computer interaction via language is the future of computers, our intersection of language and computation will play a more and more central role in our society. It is important for linguistics to invest some time now in deciding how we'd like our curriculum to reflect these changes.

Acknowledgments

Thanks to the Center for Advanced Study and to Adele Goldberg and Jerry Morgan at the University of Illinois at Urbana-Champaign, and to Alan Bell, Michelle Gregory, Lise Menn, and William Raymond for comments on this talk. Thanks to Martin Kay and Paul Smolensky for their time in discussing the history of unification and optimality. Of course any remaining errors are my own.

REFERENCES

BELL, Alan, Daniel JURAFSKY, Eric FOSLER-LUSSIER, Cynthia GIRAND, & Daniel GILDEA. 1999. Forms of English function words – Effects of disfluencies, turn position, age and sex, and predictability. To appear in *Proceedings of International Conference on Phonetic Sciences*-99.

BRENT, Michael, & Timothy A. CARTWRIGHT. 1996. Distributional regularity and phonotactic constraints are useful for segmentation. *Cognition* 61:93-126.

GILDEA, Daniel, & Daniel JURAFSKY. Learning bias and phonological rule Induction. *Computational Linguistics* 22:4.

GODFREY, J, E HOLLIMAN, & J. MCDANIEL. 1992. SWITCHBOARD: Telephone speech corpus for research and development. *Proceedings of the IEE International Conference on Acoustics, Speech, and Signal Processing*, 517-20.

GREENBERG, Steven, Dan ELLIS, & Joy HOLLENBECK. 1996. Insights into spoken language gleaned from phonetic transcription of the Switchboard corpus. *Proceedings of the Internatonal Conference on Spoken Language Processing*-96.

GREGORY, Michelle & Laura MICHAELIS. 1998. Topicalization and left-dislocation: A functional opposition revisited. Talk presented at the annual meeting of the Society for Text and Discourse, Madison Wisconsin, August 1998.

JURAFSKY, Daniel, Alan BELL, Eric FOSLER-LUSSIER, Cynthia GIRAND, & William RAYMOND. 1998. Reduction of English function words in Switchboard. *Proceedings of the Internatonal Conference on Spoken Language Processing*, 7.3111-4

——, Elizabeth SHRIBERG, Barbara FOX, & Traci CURL. 1998b. Lexical, prosodic, and syntactic cues for dialog acts. In *Joint Proceedings of the 17th International Conference on Computational Linguistics and the 35th Meeting of the Association for Computational Linguistics, Workshop on Discourse Relations and Discourse Markers.*

——, Elizabeth SHRIBERG, & Debra BIASCA. 1997. Switchboard SWBD-DMSL Labeling Project Coder's Manual, Draft 13. *University of Colorado Institute of Cognitive Science Technical Report* 97-02. Also available as http://www.colorado.edu/linguistics/jurafsky/manual.august1.html

KAY, Martin. 1979. Functional grammar. *Proceedings of5th Annual Conference of the Berkeley Linguistic Society* 5.142-58.

LANDAUER, Thomas K., & Susan T. DUMAIS. 1997. A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review* 104.211-40.

LUND, Kevin, Curt BURGESS, & R. A. ATCHLEY. 1995. Semantic and associative priming in high-dimensional semantic space. *Proceedings ofthe 17th Annual Conference of the Cognitive Science Society*, 660-5.

MACWHINNEY, Brian. 1995. The CHILDES Project. Hillsdale, NJ: Lawrence Erlbaum.

MARCUS, Mitchell, Beatrice SANTORINI, & Mary Ann MARCINKIEWICZ. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics* 19:2.313-30.

METEER, Marie, & al. 1995. *Disfluency Annotation Stylebook for the Switchboard Corpus.* Linguistic Data Consortium. Available as ftp://ftp.cis.upenn.edu/pub/treebank/swbd/doc/DFL-book.ps.gz

REGIER, Terry. 1996. *The Human Semantic Potential.* Cambridge, MA: MIT Press

SAFFRAN, Jenny R., Elissa L. NEWPORT, & Richard N. ASLIN. 1996. Word segmentation: The role of distributional cues. *Journal of Memory and Language* 35:4.606ff.

SHRIBERG, Elizabeth. 1994. Preliminaries to a theory of speech disfluencies. University of California, Berkeley, Ph.D. dissertation.

STOLCKE, Andreas. 1994. Bayesian learning of probabilistic language models. University of California, Berkeley, Ph.D. dissertation.