Psychocomputational Linguistics: A Gateway to the Computational Linguistics Curriculum

William Gregory Sakas
Department of Computer Science, Hunter College
Ph.D. Programs in Linguistics and Computer Science, The Graduate Center
City University of New York (CUNY)
695 Park Avenue, North 1008
New York, NY, USA, 10065
sakas@hunter.cuny.edu

Abstract

Computational modeling of human language processes is a small but growing subfield of computational linguistics. This paper describes a course that makes use of recent research in psychocomputational modeling as a framework to introduce a number of mainstream computational linguistics concepts to an audience of linguistics, cognitive science and computer science doctoral students. The emphasis on what I take to be the largely interdisciplinary nature of computational linguistics is particularly germane for the computer science students. Since 2002 the course has been taught three times under the auspices of the MA/PhD program in Linguistics at The City University of New York's Graduate Center. A brief description of some of the students' experiences after having taken the course is also provided.

1 Introduction

A relatively small (but growing) subfield of computational linguistics, psychocomputational modeling affords a strong foundation from which to introduce graduate students in linguistics to various computational techniques, and students in computer science[1] (*CS*) to a variety of topics in psycholinguistics, though it has rarely been incorporated into the computational linguistics curriculum.

Psychocomputational modeling involves the construction of computer models that embody one or more psycholinguistic theories of natural (human) language processing and use. Over the past decade or so, there's been renewed interest within the computational linguistics community related to the possibility of incorporating human language *strategies* into computational language *technologies*. This is evidenced by the occassional special session at computational linguistics meetings (e.g., ACL-1999 Thematic Session on Computational Psycholinguistics), several workshops (e.g., COLING-2004, ACL-2005 Psychocomputational Models of Human Language Acquisition, ACL-2004 Incremental Parsing: Bringing Engineering and Cognition Together), recent conference themes (e.g., CoNLL-2008 "... models that explain natural phenomena relating to human language") and regular invitations to psycholinguists to deliver plenary addresses at recent ACL and COLING meetings.

Unfortunately, it is too often the case that computational linguistics programs (primarily those housed in computer science departments) delay the introduction of cross-disciplinary psycholinguistic / computational linguistics approaches until either late in a student's course of study (usually as an elective) or not at all. At the City University of New York (CUNY)'s Graduate Center (the primary Ph.D.-granting school of the university) I have created a course that presents

[1] For rhetorical reasons I will often crudely partition the student makeup of the course into linguistics students and CS students. This preempts lengthy illocutions such as " … the students with a strong computational background as compared to students with a strong linguistics background." In fact there have been students from other academic disciplines in attendance bringing with them a range of technical facility in both CS and linguistics; linguistics students with an undergraduate degree in CS; and CS students with a strong undergraduate background in theoretical linguistics.

Proceedings of the Third Workshop on Issues in Teaching Computational Linguistics (TeachCL-08), pages 120–128,
Columbus, Ohio, USA, June 2008.

research in this cross-disciplinary area relatively early in the graduate curriculum. I have also run an undergraduate version of the course at Hunter College, CUNY in the interdisciplinary Thomas Hunter Honors Program.

I contend that a course developed within the mélange of psycholinguistics and computational linguistics is not only a valuable asset in a student's repertoire of graduate experience, but can effectively be used as a springboard to introduce a variety of techniques and topics central to the broader field of CL/NLP.

2 Background

The CUNY Graduate Center (hereafter, *GC*) has doctoral programs in both computer science and linguistics. The Linguistics Program also contains two master's tracks. Closely linked to both programs, but administratively independent of either, there exists a Cognitive Science Concentration.[2] In spring of 2000, I was asked by the Cognitive Science Coordinator to create an interdisciplinary course in "computing and language" that would be attractive to linguistics, speech and hearing, computer science, philosophy, mathematics, psychology and anthropology students. One might imagine how a newly-minted Ph.D. might react to this request. Well this one, not yet realizing the potential for abuse of junior faculty (a slightly more sage me later wondered if there was a last minute sabbatical-related cancellation of some course that needed to be replaced ASAP) dove in and developed *Computational Mechanisms of Syntax Acquisition.*

The course was designed to cover generative and non-generative linguistics and debates surrounding (child) first language acquisition principally focused on the question of Chomsky's Universal Grammar (*UG*), or not? Four computational models drawn from diverse paradigms – connectionist learning, statistical formal language induction, principles-and-parameters acquisition and acquisition in an optimality theory framework[3] – were presented and discussion of the UG-or-not debate was framed in the context of these models.

Over the past eight years, I've taught three variations of this course gradually molding the course away from a seminar format into a seminar/lecture format, dropping a large chunk of the UG-or-not debate, and targeting the course primarily for students who are in search of a "taste" of computational linguistics who might very well go on to take other CL-related course work.[4] What follows is a description of the design considerations, student makeup, and course content focusing on its last instantiation in Spring 2007.

3 The Course: *Computational Natural Language Learning*

Most readers will recognize the most recent title of the course which was shamelessly borrowed from the ACL's Special Interest Group on Natural Language Learning's annual meeting. Although largely an NLP-oriented meeting, the title and indeed many of the themes of the meeting's CFPs over the years accurately portray the material covered in the course.

The course is currently housed in the GC's Linguistics Program and is primarily designed to serve linguistics doctoral and masters students who want some exposure to computational linguistics but with a decidedly linguistics emphasis. Importantly though, the course often needs to serve a high percentage of students from other graduate programs.

The GC Linguistics and Computer Science Programs also offer other computational linguistics (*CL*) courses: a Natural Language Processing (*NLP*) applications survey course, a corpus analysis course, a statistical NLP course and a CL methods sequence (in addition to a small variety of electives). Although (at least until recently, see Section 7) these courses are not taught within a structured CL curriculum, they effectively serve as the "meat-and-potatoes" CL courses which require projects and assignments involving programming, a considerable math component and extensive experimentation with existing NLP/CL

[2] This is not a state-registered program, but rather an "in-house" means of allowing students to receive recognition of interdisciplinary studiy in cognitive science.

[3] Although the semester ended as we were only just getting to cover optimality theoretic acquisition.

[4] Though the details of the undergraduate version of the course are beyond the scope of this paper, it is worth noting that it did not undergo this gradual revision; it was structured much as the original Cognitive Science version of the course, actually with an increased UG-or-not discussion.

applications. The students taking these classes have already reached the point where they intend to include a substantial amount of computational linguistics in their dissertation or master's thesis.

Computational Natural Language Learning is somewhat removed from these other courses and the design considerations were purposefully directed at providing an "appetizer" that would both entice interested students into taking other courses, and prepare them with some experience in computational linguistics techniques. Over time the course has evolved to incorporate the following set of prerequisites and goals.

• *No programming prerequisite, no introduction to programming* Many of the students who take the course are first or second year linguistics students who have had little or no programming background. Students are aware that "Programming for Linguists" is part of the CL methods sequence. They come to this course looking for either an overview of CL, or for how CL concepts might be relevant to psycholinguistic or theoretical linguistics research.

Often there are students who *have* had a substantial programming background – including graduate students in computer science. This hasn't proved to be problematic since the assignments and projects are designed not to involve programming.

• *Slight math prerequisite, exposure to probabilities, and information theory* Students are expected to be comfortable with basic algebra. I dissuade students from taking the course who are intimidated by a one-line formula with a few Greek letters in it. Students are not expected to know what a conditional probability is, but will leave the course with a decent grasp of basic (undergraduate-level) concepts in probability and information theory.

This lightweight math prerequisite actually does split the class for a brief time during the semester as the probability/information theory lecture and assignment is a cinch for the CS students, and typically quite difficult for the linguistics students. But this is balanced by the implementation of the design consideration expressed in the next bullet.

• *No linguistics prerequisite, exposure to syntactic theory* Students need to know what a syntax tree is (at least in the formal language sense) but do not need to know a particular theory of human language syntax (e.g., X-bar theory or even S $\rightarrow$ NP VP). By the end of the semester students will be comfortable with elementary syntax beyond the level covered by most undergraduate "Ling 101" courses.

• *Preparation for follow-up CL courses* Students leaving this course should be comfortably prepared to enter the other GC computational linguistics offerings.[5]

• *Appreciation of the interdisciplinary nature of CL* Not all students move on to other computational linguistics courses. Perhaps the most important goal of the course is to expose CS and linguistics students (and others) to the role that computational linguistics can play in areas of theoretical linguistics and cognitive science research, and conversely to the role that cognitive science and linguistics can play in the field of computational linguistics.

3.1 Topical units

In this section I present the syllabus of the course framed in topical units. They have varied over the years; what follows is the content of the course mostly as it was taught in Spring 2007.

Janet Dean Fodor and I lead an active psychocomputational modeling research group at the City University of New York: *CUNY CoLAG* – CUNY Computational Language Acquisition Group which is primarily dedicated to the design and evaluation of computational models of first language acquisition. Most, though not all, of the topical units purposefully contain material that intersects with CUNY CoLAG's ongoing research efforts.

The depth of coverage of the units is designed to give the students some fluency in computational *issues* (e.g., use and ramifications of Markov assumptions), and a basic understanding beyond exposure to the computational *mechanisms* of CL (e.g., derivation of the standard MLE bigram

[5] The one exception in the Linguistics Program is Corpus Linguistics which has a programming prerequisite, and the occasional CL elective course in Computer Science targeted primarily for their more advanced students.

probability formula), but not designed to allow students to bypass a more computationally rigorous NLP survey course. The same is true of the breadth of coverage; a comprehensive survey is not the goal. For example, in the ngram unit, typically no more than two or at most three smoothing techniques are covered.

Note that the citations in this section are mostly required reading, but some articles are optional. It has been my experience however, that the students by and large read most of the material since the readings were highly directed (i.e., which sections and pages are most relevant to the course.) Supplemental materials that present introductory mathematics and tutorial presentations are not exhaustively listed, but included Jurafsky and Martin (2000), Goldsmith (2007, previously online) and a host of (other) online resources.

History of linguistics and computation [1 lecture] The history is framed around the question "Is computational linguistics, well uh, linguistics?" We conclude with "It was, then it wasn't, now maybe it is, or at least in part, should be." The material is tightly tied to Lee (2004); with additional discussion along the lines of Sakas (2004).

Syntax [1 lecture] This is a crash course in syntax using a context-free grammar with transformational movement. The more difficult topics include topicalization (including null-topic), Wh-movement and verb-second phenomena. We make effective use of an in-house database of abstract though linguistically viable cross-linguistic sentence patterns and tree structures – the CUNY CoLAG Domain (Sakas, 2003). The point of this lecture is to introduce non-linguistics students to the intricacies of a linguistically viable grammatical theory.

Language Acquisition [1 lecture] We discuss some of the current debates in *L1* (a child's first) language acquisition surrounding "no negative evidence" (Marcus, 1993), Poverty of the Stimulus (Pullum, 1996), and Chomsky's conceptualization of Universal Grammar. This is the least computational lecture of the semester, although it often generates some of the most energized discussion. The language acquisition unit is the central arena in which we stage most of the rest of the topics in the course.

Gold and the Subset Principle [2 lectures] During the presentation of Gold's (1967) and Angluin's (1980) proofs and discussion of how they might be used to argue (often incorrectly) for a Universal Grammar (Johnson, 2004) some core CL topics are introduced including formal language classes (the Chomsky Hierarchy) and the notions of hypothesis space and search space. The first (toy) probabilistic analyses are also presented (e.g., given a finite enumeration and a probability *p* that an arbitrary non-target grammar licenses a sentence in the input sample, what is the "worst case" number of sentences required to converge on the target grammar?)

Next, the Subset Principle and linguistic overgeneralization (Fodor and Sakas, 2005) is introduced. An important focus is on how keeping (statistical) track of what's *not* encountered might supply a 'retreat' mechanism to pull back from an over-general hypothesis. Although the mathematical details of how the statistics might work are omitted, this topic leads neatly into a unit on Bayesian learning later in the semester.

This is an important two lectures. It's the first unit where students are exposed to the use of computational techniques applied to theoretical issues in psycholinguistics. By this point, students often are intellectually engaged in the debates surrounding L1 acquisition. To understand the arguments presented in this unit students need to flex their computational muscles for the first time.

Connectionism [3 lectures] This unit covers the basics of Simple Recurrent Network (SRN) learning (Elman, 1990, 1993). More or less, Elman argues that language acquisition is not necessarily the acquisition of rules operating over atomic linguistic units (e.g., phrase markers) but rather the process of capturing the "dynamics" of word patterns in the input stream. He demonstrates how this can be simulated in an SRN paradigm.

The mechanics of how an SRN operates and can be used to model language acquisition phenomena is covered but more importantly core concepts common to most all supervised machine learning paradigms are emphasized. Topics include how training and testing corpora are developed and used, cross validation, hill-climbing, learning bias, linear and non-linear separation of the hypothesis space, etc. A critique of SRN learning is also covered (Marcus, 1998) which presents the important distinction between *generalization performance* and *learning within the training space* in a way that is approachable by non-CS students, but also interesting to CS-students.

Information retrieval [1 lecture] Elman (1990) uses hierarchal clustering to analyze some of his results. I use Elman's application of clustering to take a brief digression from the psycholinguistics theme of the course and present an introduction to vector space models and document clustering.

This is the most challenging technical lecture of the semester and is included only when there are a relatively high proportion of CS students in attendance. Most of the linguistics students get a decent feel for the material, but most require a second exposure it in another course to fully understand the math. That said, the linguistics students do understand how weight heuristics are used to effectively represent documents in vectors (though most linguistics students have a hard time swallowing the bag-of-words paradigm at face value), and how vectors can be nearer or farther from each other in a hypothesis space.

Ngram language models [3 lecture] In this unit we return to psycholinguistics. Reali and Christiansen, (2005) present a simple ngram language model of child-directed speech to argue against the need for innate UG-provided knowledge of hierarchal syntactic structure. Basic probability and information theory is introduced – conditional probabilities and Markov assumptions, the chain rule, Bayes Rule, maximum likelihood estimates, entropy, etc. Although relatively easy for the CS students (they had their hands full with the syntax unit), introduction of this material is invaluable to the linguistics students who need to be somewhat fluent in it before entering our other CL offerings.

We continue with a presentation of the sparse data problem, Zipf’s law, corpus cross-entropy and a handful of smoothing techniques (Reali & Christiansen use a particularly impoverished version of deleted interpolation). We continue with a discussion of the pros and cons of employing Markov assumptions in computational linguistics generally, the relationship of Markov assumptions to incremental learning and psycholinguistic modeling, and the use of cross-entropy as an evaluation metric, and end with a brief discussion of the descriptional necessity (or not) of traditional generative grammars (Pereira, 2000).

.

"Ideal" learning, Bayesian learning and computational resources [1 lecture] Regier and Gahl (2004) in response to Lidz et al. (2003) present a Bayesian learning model that learns the correct structural referent for anaphoric "one" in English from a corpus of child-directed speech. Similarly to Reali & Christiansen (op. cit.), they argue against the need for innate knowledge of hierarchal structure since their Bayesian model starts tabula rasa and learns from linear word strings with no readily observable structure.

The fundamental mechanics of Bayesian inference is presented. Since most Bayesian models are able to retreat from overgeneral hypotheses in the absence of positive evidence, the course returns to overgeneralization errors, the Subset Principle and the alternative of using statistics as a possible retreat mechanism. Computationally heavy ("ideal") batch processing, and incremental (psychologically plausible) processing are contrasted here as is the use of heuristics (psycholinguistically-based or not) to mitigate the potentially huge computational cost of searching a large domain.

Principle and parameters [2 lectures] As the academic scheduling has worked out, the course is usually offered during years when the Linguistics Program does not offer a linguistics-based learnability course. As a result, there is a unit on acquisition within a principles-and-parameters (*P&P*) framework (Chomsky, 1981). Roughly, in the P&P framework cross-linguistic commonalities are considered principles, and language variation is standardly specified by the settings of a bank of binary parameters (i.e., UG = principles + parameters; a specific language = principles + parameters + parameter settings).

Although this unit is the furthest away from mainstream CL, it has served as a useful means to introduce deterministic learning (Sakas and Fodor, 2001), versus non-deterministic learning (Yang, 2002), the effectiveness of hill-climbing in linguistically smooth and non-smooth domains,[6] as well as the notion of computational complexity and combinatorial explosion (*n* binary parameters yields a search space of 2^n possible grammars). Finally, and perhaps most importantly there is extensive discussion of the difficulty of building computational systems that can efficiently and correctly learn to navigate through domains with an enormous amount of ambiguity.

In the P&P framework ambiguity stems from competition of cross-linguistic structural analyses of surface word order patterns. For example, given a (tensed) *S V O* sentence pattern, is the *V* situated under the phrase maker *I* (English), or under the phrase marker *C* (German)? Although this is a somewhat different form of ambiguity than the within-language structural ambiguity that is all too familiar to those of use working in CL, it serves as useful background material for the next unit.

Part of speech tagging and statistical parsing [3 lectures] In this unit we begin by putting aside the psycholinguistics umbrella of the course and cover introductory CL in a more traditional manner. Using Charniak (1997) as the primary reading, we cover rudimentary HMM's, and probabilistic CFG's. We use supplemental materials to introduce lexicalized statistical parsing (e.g., Jurafsky and Martin, 2000 and online materials). We then turn back to psycholinguistics and after a (somewhat condensed overview) of human sentence processing, discuss the viability of probabilistic parsing as a model of human sentence processing (Keller, 2005). This unit, more than some others, is lightweight on detailed computational mechanics; the material is presented throughout at a level similar to that of Charniak’s article. For example the specifics of EM algorithms are not covered although what they do, and why they are necessary are.

The Linguistics Program at CUNY is very active in human sentence processing research and this unit is of interest to many of the linguistics students. In particular we contrast computational approaches that employ nondeterminism and parallelism to mainstream psycholinguistics models which are primarily deterministic, serial and employ a reanalysis strategy when evaluating a parse “online” (though of course there is a significant amount of research that falls outside of this mainstream). We then focus on issues of computational resources that each paradigm requires.

In some ways the last lectures of this unit best embody the goal of exposing the students to the potential of interdisciplinary research in computational linguistics. The CS students leave with an appreciation of psycholinguistic approaches to human sentence processing, and the linguistics students with a firm grasp of the effectiveness of computational approaches.

4 Assignments and Projects

Across the three most recent incarnations of the course the number and difficulty of the assignments and projects has varied quite a bit. In the last version, there were three assignments (five to ten hours of student effort each) and one project (twenty to thirty hours effort).

Due to the typically small size of the course, assignments and projects (beyond weekly readings) were often individually tailored and assessed. The goal of the assignments was to concretize the CL aspects of the primarily psycholinguistic readings with either hands-on use of the computer, mathematically-oriented problem sets, or a critical evaluation of the CL methodologies employed. A handful of examples follow.

- **Gold and the Subset Principle (Assignments)** All students are asked to formulate a Markov chain (though at this point in the course, not by that name) of a Gold-style enumeration learner operating over a small finite domain (e.g., 4 grammars, 12 sentences and a sentence to grammar mapping). The more mathematically inclined are additionally asked to calculate the expected value of the number of input sentences consumed by a learner operating over an enumeration of *n* grammars and given a generalized mapping of sentences to grammars, or to formally prove the learnability of any finite domain of languages given text (positive) presentation of input.

- **Connectionism (Assignments)** All students were asked to pick a language from the CUNY CoLAG domain, develop a training and test set

[6] By "smooth", I mean a correlation between the similarity of grammars, and the similarity of languages they generate.

from that language using existing software and run a predict-the-next-word SRN simulation on either a MatLab or TLearn neural network platform. Linguistics and CS students were paired on this assignment. When the assignment is given, a relatively detailed after-class lab tutorial on how to run the software is presented.

• **Ngram language models (Projects)** One CS student implemented a version of Reali and Christiansen's experiment and was asked to evaluate the effectiveness of different smoothing techniques on child-directed speech and to design a study of how to evaluate differences between child-directed speech and adult-to-adult speech in terms of language modeling. A linguistics student was asked to write a paper explaining how one could develop a computational evaluation of how a bigram learner might be evaluated longitudinally. (I.e., to answer the question, how can one measure the effectiveness of a language model after each input sentence?). Another linguistics student (with strong CS skills) created an annotation tool that semi-automatically mapped child-directed speech in French onto the CoLAG Domain tag set.

5 Students: Past and Current

As mentioned earlier, the Linguistics Doctoral Program at CUNY has just recently begun to structure their computational linguistics offerings into a cohesive course of study (described briefly in Section 7). During the past several years Computational Natural Learning has been offered on an ad hoc basis primarily in response to student demand and demographics of students' computational skills. Since the course was not originally intended to serve any specific function as part of a larger curriculum, and was not integrated into a reoccurring schedule there has been little need to carry out a systematic evaluation of the impact of the course on students' academic careers. Still a few anecdotal accounts will help give a picture of the course's effectiveness.

After the first Cognitive Science offering of the course in 2000, approximately 30 graduate students have taken one of the three subsequent incarnations. Two of the earliest linguistics students went on to take undergraduate CS courses in programming and statistics, and subsequently came back to take graduate level CL courses.[7] They have obtained their doctorates and are currently working in industry as computational linguists. One is a lead software engineer for an information retrieval startup company in New York that does email data mining. And though I've lost track of the other student, she was at one point working for a large software company on the west coast.

I am currently the advisor of one CS student, and two linguistics students who have taken the course. One linguistics student is in the throws of writing a dissertation on the plausibility of exploiting statistical regularities of various syntactic structures (contra regularities of word strings) in child-directed speech during L1 acquisition. The other is relatively early in her academic career, but is interested in working on computational semantics and discourse analysis within a categorial grammar framework. Her thoughts currently revolve around establishing (and formalizing) relationships between traditional linguistics-oriented semantics and a computational semantics paradigm. She hopes to make contributions to both linguistics and CL. The CS student, also early in his career, is interested in semi-productive multi-word expressions and how young children can come to acquire them. His idea is to employ a learning component in a machine translation system that can be trained to translate productive metaphors between a variety of languages.

These five students have chosen to pursue specific areas of study and research directly as a result of having taken Computational Natural Language Learning early in their careers.

I am also sitting on two CS students' second qualifying exam committees. One is working on machine translation of Hebrew and the other working on (relatively) mainstream word-sense disambiguation. Both of their qualifying exam papers show a sensitivity to psycholinguistics that I'm frankly quite happy to see, and am sure wouldn't have been present without their having taken the course.

The parsing unit was just added this past spring semester and I've had two recent discussions with

[7] The CL methods sequence was established only 3 years ago, previously students were encouraged to develop their basic computational skills at one of CUNY's undergraduate schools.

a second year linguistics student about incorporating a statistical component into a current psycholinguistic model of human sentence processing. Another second year student has expressed interested in doing a comprehensive study of neural network models of human sentence parsing for his first qualifying paper. It's not clear that they will ultimately pursue these directions, but I'm certain they wouldn't have thought of the possibilities if they hadn't taken the Computational Natural Language Learning.

Finally, most all of the students who have taken the course have also taken the NLP-survey course (no programming required), slightly less than a third have moved on to the CL methods sequence (includes an introduction to programming), or if they have some CS background move directly to Corpus Analysis (programming experience required as a prerequisite). We hope that eventually, especially in light of the GC's new computational linguistics program, the course will serve as the gateway for many more students to begin to pursue studies that will lead to research areas in both psychocomputational modeling and more mainstream CL.

6 Brief Discussion

It is my view that computational linguistics is by nature a cross-disciplinary endeavor. Indeed, one could argue that only after the incorporation of techniques and strategies gleaned from theoretical advances in psychocomputational modeling of language, can we achieve truly transparent (to the user) human-computer language applications.

That argument notwithstanding, a course such as the one described in this paper can effectively serve as an introduction to an assortment of concepts in computational linguistics that can broaden the intellectual horizons of both CS and linguistics students, as well providing a foundation that students can build on in the pursuit of more advanced studies in the field.

7 Postscript: The Future of the Course

The CUNY Graduate Center has recently created a structured computational linguistics program housed in the GC's Linguistics Program. The program consists of a Computational Linguistics Concentration in the Linguistics Master's subprogram, and particularly relevant to the discussion in this article, a Computational Linguistics Certificate[8] (both fall under the acronym *CLC*). Any City University of New York doctoral student can enroll in CLC concurrently with enrollment in their primary doctoral program (as one might imagine, we expect a substantial number of Linguistics and CS doctoral candidates to enroll in the CLC program).

Due to my newly-acquired duties as director of the CLC program and to scheduling constraints on CLC faculty teaching assignments, the course cannot be offered again until Fall 2009 or Spring 2010. At that time Computational Natural Language Learning will need to morph into a more technically advanced elective course in applied machine learning techniques in computational linguistics (or some such) since the CLC course of study currently posits the NLP survey course and the CL Methods sequence as the first year introductory requirements.

However, I expect that a course similar to the one described here will supplement the NLP survey course as a first year requirement in Fall 2010. The course will be billed as having broad appeal and made available to both CLC students *and* linguistics, CS and other students who might not want or require the "meat-and-potatoes" that CLC offers, but who only desire a CL "appetizer". Though if the appetizer is tasty enough, students may well hunger for the main course.

Acknowledgments

I would like to thank the three anonymous reviewers for helpful comments, and the many intellectually diverse and engaging students I've had the pleasure to introduce to the field of computational linguistics.

References

Angluin, D. (1980). Inductive inference of formal languages from positive data. *Information and Control* 45:117-135.

Charniak, E. (1997). Statistical Techniques for Natural language Parsing. *AI Magazine* 18:33-44.

Chomsky, N. (1981). *Lectures on government and binding*: Studies in generative grammar. Dordrecht: Foris.

[8] Pending state Department of Education approval, hopefully to be received in Spring 2009.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science* 14:179-211.

Elman, J. L. (1993). Learning and development in neural networks: The importance of starting small. *Cognition* 48:71-99.

Fodor, J. D., and Sakas, W. G. (2005). The Subset Principle in syntax: Costs of compliance. *Journal of Linguistics* 41:513-569.

Gold, E. M. (1967). Language identification in the limit. *Information and Control* 10:447-474.

Goldsmith, J. (2007). Probability for Linguists. *Mathematics and Social Sciences* 180:5-40.

Johnson, K. (2004). Gold's Theorem and Cognitive Science. *Philosophy of Science* 71:571-592.

Jurafsky, D., and Martin, J. H. (2000). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*

Keller, F. (2005). Probabilistic Models of Human Sentence Processing. Presented at *Probabilistic Models of Cognition: The Mathematics of Mind*, IPAM workshop, Los Angeles.

Lee, L. (2004). "I'm sorry Dave, I'm afraid I can't do that" : Linguistics, statistics, and natural language processing circa 2001. In *Computer Science: Reflections on the Field*, 111-118. Washington:National Academies Press.

Lidz, J., Waxman, S., and Freedman, J. (2003). What infants know about syntax but couldn't have learned: Experimental evidence for syntactic structure at 18 months. *Cognition* 89:65-73.

Marcus, G. F. (1993). Negative evidence in language acquisition. *Cognition* 46:53-85.

Marcus, G. F. (1998). Can connectionism save constructivism? *Cognition* 66:153-182.

Pereira, F. (2000) Formal grammar and information theory: Together again. *Philosophical Transactions of the Royal Society* A358:1239-1253.

Pullum, G. K. (1996). Learnability, hyperlearning, and the poverty of the stimulus. *Proceedings of the 22nd Annual Meeting of the Berkley Linguistics Society: General Session and Parasession on the Role of Learnability in Grammatical Theory*, Berkeley: 498-513.

Reali, F., and Christiansen, M. H. (2005). Uncovering the richness of the stimulus: Structural dependence and indirect statistical evidence. *Cognitive Science* 29:1007-10018.

Regier, T., and Gahl, S. (2004). Learning the unlearnable: The role of missing evidence. *Cognition* 93:147-155.

Sakas, W. G., and Fodor, J. D. (2001). The Structural Triggers Learner. In *Language Acquisition and Learnability*, ed. S. Bertolo, 172-233. Cambridge: Cambridge University Press.

Sakas, W. G. (2003) A Word-Order Database for Testing Computational Models of Language Acquisition, *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, ACL-2003: 415-422.

Sakas, W. G. (2004) Introduction. *Proceedings of the First Workshop on Psycho-computational Models of Human Language Acquisition*, COLING-2004. Geneva: iv-vi.

Yang, C. D. (2002). *Knowledge and learning in natural language*. New York: Oxford University Press.