How the statistical revolution changes (computational) linguistics

Mark Johnson
Cognitive and Linguistic Sciences and Computer Science
Brown University
Mark_Johnson@Brown.edu

Abstract

This paper discusses some of the ways that the "statistical revolution" has changed and continues to change the relationship between linguistics and computational linguistics. I claim that it is more useful in parsing to make an open world assumption about possible linguistic structures, rather than the closed world assumption usually made in grammar-based approaches to parsing, and I sketch two different ways in which grammar-based approaches might be modified to achieve this. I also describe some of the ways in which probabilistic models are starting to have a significant impact on psycholinguistics and language acquisition. In language acquisition Bayesian techniques may let us empirically evaluate the role of putative universals in universal grammar.

1 Introduction

The workshop organizers asked us to write something controversial to stimulate discussion, and I've attempted to do that here. Usually in my papers I try to stick to facts and claims that I can support, but here I have fearlessly and perhaps foolishly gone out on a limb and presented guesses, hunches and opinions. Take them with a grain of salt. Inspired by Wanamaker's well-known quote about advertising, I expect that half of the ideas I'm proposing here are wrong, but I don't know which half. I hope the conference will help me figure that out.

Statistical techniques have revolutionized many scientific fields in the past two decades, including computational linguistics. This paper discusses the impact of this on the relationship between computational linguistics and linguistics. I'm presenting a personal perspective rather than a scientific review here, and for this reason I focus on areas I have some experience with. I begin by discussing how the statistical perspective changed my understanding of the relationship between linguistic theory, grammars and parsing, and then go on to describe some of the ways that ideas from statistics and machine learning are starting to have an impact on linguistics today.

Before beginning, I'd like to say something about what I think computational linguistics is. I view computational linguistics as having both a scientific and an engineering side. The engineering side of computational linguistics, often called natural language processing (NLP), is largely concerned with building computational tools that do useful things with language, e.g., machine translation, summarization, question-answering, etc. Like any engineering discipline, natural language processing draws on a variety of different scientific disciplines.

I think it's fair to say that in the current state of the art, natural language processing draws far more heavily on statistics and machine learning than it does on linguistic theory. For example, one might claim that all an NLP engineer really needs to understand about linguistic theory are (say) the parts of speech (POS). Assuming this is true (I'm not sure it is), would it indicate that there is something wrong with either linguistic theory or computational linguistics? I don't think it does: there's no reason to expect an engineering solution to utilize all the scientific knowledge of a related field. The fact that you can build perfectly good bridges with Newtonian mechanics says nothing about the truth of quantum mechanics.

I also believe that there is a scientific field of computational linguistics. This scientific field exists not just because computers are incredibly useful for doing linguistics — I expect that computers have revolutionized most fields of science — but because it makes sense to think of linguis-

Proceedings of the EACL 2009 Workshop on the Interaction between Linguistics and Computational Linguistics, pages 3–11,
Athens, Greece, 30 March, 2009. ©2009 Association for Computational Linguistics

tic *processes* as being essentially computational in nature. If we take computation to be the manipulation of symbols in a meaning-respecting way, then it seems reasonable to hypothesize that language comprehension, production and acquisition are all computational processes. Viewed this way, we might expect computational linguistics to interact most strongly with those areas of linguistics that study linguistic processing, namely psycholinguistics and language acquisition. As I explain in section 3 below, I think we are starting to see this happen.

2 Grammar-based and statistical parsing

In some ways the 1980s were a golden age for collaboration and cross-fertilization between linguistic theory and computational linguistics, especially between syntax and parsing. Gazdar and colleagues showed that Chomskyian transformations could be supplanted by computationally much simpler feature passing mechanisms (Gazdar et al., 1985), and this lead to an explosion of work on "unification-based" grammars (Shieber, 1986), including the Lexical-Functional Grammars and Head-driven Phrase Structure Grammars that are still very actively pursued today. I'll call the work on parsing within this general framework the *grammar-based approach* in order to contrast it with the *statistical approach* that doesn't rely on these kinds of grammars. I think the statistical approach has come to dominate computational linguistics, and in this section I'll describe why this happened.

Before beginning I think it's useful to clarify our goals for building parsers. There are many reasons why one might build any computational system — perhaps it's a part of a commercial product we hope will make us rich, or perhaps we want to test the predictions of a certain theory of processing — and these reasons should dictate how and even whether the system is constructed. I'm assuming in this section that we want to build parsers because we expect the representations they produce will be useful for various other NLP engineering tasks. This means that parser design is itself essentially an engineering task, i.e., we want a device that returns parses that are accurate as possible for as many sentences as possible.

I'll begin by discussing a couple of differences between the approaches that are often mentioned but I don't think are really that important. The grammar-based approaches are sometimes described as producing deeper representations that are closer to meaning. It certainly is true that grammar-based analyses typically represent predicate-argument structure and perhaps also quantifier scope. But one can recover predicate-argument structure using statistical methods (see the work on semantic role labeling and "PropBank" parsing (Palmer et al., 2005)), and presumably similar methods could be used to resolve quantifier scope as well.

I suspect the main reason why statistical parsing has concentrated on more superficial syntactic structure (such as phrase structure) is because there aren't many actual applications for the syntactic analyses our parsers return. Given the current state-of-the-art in knowledge representation and artificial intelligence, even if we could produce completely accurate logical forms in some higher-order logic, it's not clear whether we could do anything useful with them. It's hard to find real applications that benefit from even syntactic information, and the information any such applications actually use is often fairly superficial. For example, some research systems for named entity detection and extraction use parsing to identify noun phrases (which are potentially name entities) as well as the verbs that govern them, but they ignore the rest of the syntactic structure. In fact, many applications of statistical parsers simply use them as language models, i.e., one parses to obtain the probability that the parser assigns to the string and throws away the parses it computes in the process (Jelinek, 2004). (It seems that such parsing-based language models are good at preferring strings that are at least superficially grammatical, e.g., where each clause contains one verb phrase, which is useful in applications such as summarization and machine translation).

Grammar-based approaches are also often described as more linguistically based, while statistical approaches are viewed as less linguistically informed. I think this view primarily reflects the origins of the two approaches: the grammar-based approach arose from the collaboration between linguists and computer scientists in the 1980s mentioned earlier, while the statistical approach has its origins in engineering work in speech recognition in which linguists did not play a major role. I also think this view is basically false. In the grammar-based approaches linguists write the grammars while in statistical approaches linguists annotate the corpora with syntactic parses, so linguists play a central role in both. (It's an interesting question as to why corpus annotation plus statistical inference seems to be a more effective way of getting linguistic information into a computer than manually writing a grammar).

Rather, I think that computational linguists working on statistical parsing need a greater level of linguistic sensitivity at an informal level than those working on grammar-based approaches. In the grammar-based approaches all linguistic knowledge is contained in the grammar, which the computational linguist implementing the parsing framework doesn't actually have to understand. All she has to do is correctly implement an inference engine for grammars written in the relevant grammar formalism. By contrast, statistical parsers define the probability of a parse in terms of its (statistical) features or properties, and a parser designer needs to choose which features their parser will use, and many of these features reflect at least an intuitive understanding of linguistic dependencies. For example, statistical parsers from Magerman (1995) on use features based on head-dependent relationships. (The parsers developed by the Berkeley group are a notable exception (Petrov and Klein, 2007)). While it's true that only a small fraction of our knowledge about linguistic structure winds up expressed by features in modern statistical parsers, as discussed above there's no reason to expect all of our scientific knowledge to be relevant to any engineering problem. And while many of the features used in statistical parsers don't correspond to linguistic constraints, nobody seriously claims that humans understand language only using linguistic constraints of the kind expressed in formal grammars. I suspect that many of the features that have been shown to be useful in statistical parsing encode psycholinguistic markedness preferences (e.g., attachment preferences) and at least some aspects of world knowledge (e.g., that the direct object of "eat" is likely to be a food).

Moreover, it's not necessary for a statistical model to exactly replicate a linguistic constraint in order for it to effectively capture the corresponding generalization: all that's necessary is that the statistical features "cover" the relevant examples. For example, adding a subject-verb agreement feature to the Charniak-Johnson parser (Charniak and Johnson, 2005) has no measurable effect on parsing accuracy. After doing this experiment I realized this shouldn't be surprising: the Charniak parser already conditions each argument's part-of-speech (POS) on its governor's POS, and since POS tags distinguish singular and plural nouns and verbs, these general head-argument POS features capture most cases of subject-verb agreement.

Note that I'm not claiming that subject-verb agreement isn't a real linguistic constraint or that it doesn't play an important role in human parsing. I think that the type of input (e.g., treebanks) and the kinds of abilities (e.g., to exactly count the occurences of many different constructions) available to our machines may be so different to what is available to a child that the features that work best in our parsers need not bear much relationship to those used by humans.

Still, I view the design of the features used in statistical parsers as a fundamentally linguistic issue (albeit one with computational consequences, since the search problem in parsing is largely determined by the features involved), and I expect there is still more to learn about which combinations of features are most useful for statistical parsing. My guess is that the features used in e.g., the Collins (2003) or Charniak (2000) parsers are probably close to optimal for English Penn Treebank parsing (Marcus et al., 1993), but that other features might improve parsing of other languages or even other English genres. Unfortunately changing the features used in these parsers typically involves significant reprogramming, which makes it difficult for linguists to experiment with new features. However, it might be possible to develop a kind of statistical parsing framework that makes it possible to define new features and integrate them into a statistical parser without any programming which would make it easy to explore novel combinations of statistical features; see Goodman (1998) for an interesting suggestion along these lines.

From a high-level perspective, the grammar-based approaches and the statistical approaches both view parsing fundamentally in the same way, namely as a specialized kind of inference problem. These days I view "parsing as deduction" (one of the slogans touted by the grammar-based crowd) as unnecessarily restrictive; after all, psycholinguistic research shows that humans are exquisitely sensitive to distributional information, so why shouldn't we let our parsers use that information as well? And as Abney (1997) showed, it is mathematically straight-forward to define probability distributions over the representations used by virtually any theory of grammar (even those of Chomsky's Minimalism), which means that theoretically the arsenal of statistical methods for parsing and learning can be applied to any grammar just as well.

In the late 1990s I explored these kinds of statistical models for Lexical-Functional Grammar (Bresnan, 1982; Johnson et al., 1999). The hope was that statistical features based on LFG's richer representations (specifically, f-structures) might result in better parsing accuracy. However, this seems not to be the case. As mentioned above, Abney's formulation of probabilistic models makes essentially no demands on what linguistic representations actually are; all that is required is that the statistical features are functions that map each representation to a real number. These are used to map a set of linguistic representations (say, the set of all grammatical analyses) to a set of vectors of real numbers. Then by defining a distribution over these sets of real-valued vectors we implicitly define a distribution over the corresponding linguistic representations.

This means that as far as the probabilistic model is concerned the details of the linguistic representations don't actually matter, so long as there are the right number of them and it is possible to compute the necessary real-valued vectors from them. For a computational linguist this is actually quite a liberating point of view; we aren't restricted to slavishly reproducing textbook linguistic structures, but are free to experiment with alternative representations that might have computational or other advantages.

In my case, it turned out that the kinds of features that were most useful for stochastic LFG parsing could in fact be directly computed from phrase-structure trees. The features that involved f-structure properties could be covered by other features defined directly on the phrase-structure trees. (Some of these phrase-structure features were implemented by rather nasty C++ routines but that doesn't matter; Abney-type models make no assumptions about what the feature functions are). This meant that I didn't actually need the f-structures to define the probability distributions I was interested in; all I needed were the corresponding c-structure or phrase-structure trees.

And of course there are many ways of obtaining phrase-structure trees. At the time my colleague Eugene Charniak was developing a statistical phrase-structure parser that was more robust and had broader coverage than the LFG parser I was working with, and I found I generally got better performance if I used the trees his parser produced, so that's what I did. This leads to the discriminative re-ranking approach developed by Collins and Koo (2005), in which a statistical parser trained on a treebank is used to produce a set of candidate parses which are then "re-ranked" by an Abney-style probabilistic model.

I suspect these robustness and coverage problems of grammar-based parsing are symptoms of a fundamental problem in the standard way that grammar-based parsing is understood. First, I think grammar-based approaches face a dilemma: on the one hand the explosion of ambiguity suggests that some sentences get too many parses, while the problems of coverage show that some sentences get too few, i.e., zero, parses. While it's possible that there is a single grammar that can resolve this dilemma, my point here is that each of these problems suggests we need to modify the grammars in exactly the opposite way, i.e., generally tighten the constraints in order to reduce ambiguity, while generally relax the constraints in order to allow more parses for sentences that have no parses at all.

Second, I think this dilemma only arises because the grammar-based approach to parsing is fundamentally designed around the goal of distinguishing grammatical from ungrammatical sentences. While I agree with Pullum (2007) that grammaticality is and should be central to syntactic theory, I suspect it is not helpful to view parsing (by machines or humans) as a byproduct of proving the grammaticality of a sentence. In most of the applications I can imagine, what we really want from a parser is the parse that reflects its best guess at the intended interpretation of the input, even if that input is ungrammatical. For example, given the telegraphese input "man bites dog" we want the parser to tell us that "man" is likely to be the agent of "bites" and "dog" the patient, and not simply that the sentence is ungrammatical.

These grammars typically distinguish grammatical from ungrammatical analyses by explicitly characterizing the set of grammatical analyses in some way, and then assuming that all other analyses are ungrammatical. Borrowing terminology from logic programming (Lloyd, 1987) we might call this a *closed-world assumption*: any analysis the grammar does not generate is assumed to be ungrammatical.

Interestingly, I think that the probabilistic models used statistical parsing generally make an *open-world assumption* about linguistic analyses. These probabilistic models prefer certain linguistic structures over others, but the smoothing mechanisms that these methods use ensure that every possible analysis (and hence every possible string) receives positive probability. In such an approach the statistical features identify properties of syntactic analyses which make the analysis more or less likely, so the probabilistic model can prefer, disprefer or simply be ambivalent about any particular linguistic feature or construction.

I think an open-world assumption is generally preferable as a model of syntactic parsing in both humans and machines. I think it's not reasonable to assume that the parser knows all the lexical entries and syntactic constructions of the language it is parsing. Even if the parser encounters a word or construction it doesn't understand it, that shouldn't stop it from interpreting the rest of the sentence. Statistical parsers are considerably more open-world. For example, unknown words don't present any fundamental problem for statistical parsers; in the absence of specific lexical information about a word they automatically back off to generic information about words in general.

Does the closed-world assumption inherent in the standard approach to grammar-based parsing mean we have to abandon it? I don't think so; I can imagine at least two ways in which the conventional grammar-based approach might be modified to obtain an open-world parsing model.

One possible approach keeps the standard closed-world conception that grammars generate only grammatical analyses, but gives up the idea that parsing is a byproduct of determining the grammaticality of the input sentence. Instead, we might use a *noisy channel* to map grammatical analyses generated by the grammar to the actual input sentences we have to parse. Parsing involves recovering the grammatical source or underlying sentence as well as its structure. Presumably the channel model would be designed to prefer minimal distortion, so if the input to be parsed is in fact grammatical then the channel would prefer the identity transformation, while if the input is ungrammatical the channel model would map it to close grammatical sentences. For example, if such a parser were given the input "man bites dog" it might decide that the most probable underlying sentence is "a man bites a dog" and return a parse for that sentence. Such an approach might be regarded as a way of formalizing the idea that ungrammatical sentences are interpreted by analogy with grammatical ones. (Charniak and I proposed a noisy channel model along these lines for parsing transcribed speech (Johnson and Charniak, 2004)).

Another possible approach involves modifying our interpretation of the grammar itself. We could obtain an open world model by relaxing our interpretation of some or all of the constraints in the grammar. Instead of viewing them as hard constraints that define a set of grammatical constructions, we reinterpret them as violable, probabilistic features. For example, instead of interpreting subject-verb agreement as a hard constraint that rules out certain syntactic analyses, we reinterpret it as a soft constraint that penalizes analyses in which subject-verb agreement fails. Instead of assuming that each verb comes with a fixed set of subcategorization requirements, we might view subcategorization as preferences for certain kinds of complements, implemented by features in an Abney-style statistical model. Unknown words come with no subcategorization preferences of their own, so they would inherit the prior or default preferences. Formally, I think this is fairly easy to achieve: we replace the hard unification constraints (e.g., that the subject's number feature equals the verb's number feature) with a stochastic feature that fires whenever the subject's number feature differs from the verb's number feature, and rely on the statistical model training procedure to estimate that feature's weight.

Computationally, I suspect that either of these options (or any other option that makes the grammar-based approaches open world) will require a major rethinking of the parsing process. Notice that both approaches let ambiguity proliferate (ambiguity is our friend in the fight against poor coverage), so we would need parsing algorithms capable of handling massive ambiguity. This is true of most statistical parsing models, so it is possible that the same approaches that have proven successful in statistical parsing (e.g., using probabilities to guide search, dynamic programming, coarse-to-fine) will be useful here as well.

3 Statistical models and linguistics

The previous section focused on syntactic parsing, which is an area in which there's been a fruitful interaction between linguistic theory and computational linguistics over a period of several decades. In this section I want to discuss two other emerging areas in which I expect the interaction between linguistics and computational linguistics to become increasingly important: psycholinguistics and language acquisition. I think it's no accident that these areas both study processing (rather than an area of theoretical linguistics such as syntax or semantics), since I believe that the scientific side of computational linguistics is fundamentally about such linguistic processes.

Just to be clear: psycholinguistics and language acquisition are experimental disciplines, and I don't expect the average researcher in those fields to start doing computational linguistics any time soon. However, I do think there are an emerging cadre of young researchers in both fields applying ideas and results from computational linguistics in their work and using experimental results from their field to develop and improve the computational models. For example, in psycholinguistics researchers such as Hale (2006) and Levy (2008) are using probabilistic models of syntactic structure to make predictions about human sentence processing, and Bachrach (2008) is using predictions from the Roark (2001) parser to help explain the patterns of fMRI activation observed during sentence comprehension. In the field of language acquisition computational linguists such as Klein and Manning (2004) have studied the unsupervised acquisition of syntactic structure, while linguists such as Boersma and Hayes (2001), Goldsmith (2001), Pater (2008) and Albright and Hayes (2003) are developing probabilistic models of the acquisition of phonology and/or morphology, and Frank et al. (2007) experimentally tests the predictions of a Bayesian model of lexical acquisition. Since I have more experience with computational models of language acquisition, I'll concentrate on this topic for the rest of this section.

Much of this work can be viewed under the slogan "structured statistical learning". That is, specifying the structures over which the learning algorithm generalizes is just as important as specifying the learning algorithm itself. One of the things I like about this work is that it gets beyond the naive nature-versus-nurture arguments that characterize some of the earlier theoretical work on language acquisition. Instead, these computational models become tools for investigating the effect of specific structural assumptions on the acquisition process. For example, Goldwater et al. (2007) shows that modeling inter-word dependencies improves word segmentation, which shows that the linguistic context contains information that is potentially very useful for lexical acquisition.

I think it's no accident that much of the computational work is concerned with phonology and morphology. These fields seem to be closer to the data and the structures involved seem simpler than in, say, syntax and semantics. I suspect that linguists working in phonology and morphology find it easier to understand and accept probabilistic models in large part because of Smolensky's work on Optimality Theory (Smolensky and Legendre, 2005). Smolensky found a way of introducing optimization into linguistic theory in a way that linguists could understand, and this serves as a very important bridge for them to probabilistic models.

As I argued above, it's important with any computational modeling to be clear about exactly what our computational models are intended to achieve. Perhaps the most straight-forward goal for computational models of language acquisition is to view them as specifying the actual computations that a human performs when learning a language. Under this conception we expect the computational model to describe the learning trajectory of language acquisition, e.g., if it takes the algorithm more iterations to learn one word than another, then we would expect humans to take longer to that word as well. Much of the work in computational phonology seems to take this perspective (Boersma and Hayes, 2001).

Alternatively, we might view our probabilistic models (rather than the computational procedures that implementing them) as embodying the scientific claims we want to make. Because these probabilistic models are too complex to analyze analytically in general we need a computational procedure to compute the model's predictions, but the computational procedure itself is not claimed to have any psychological reality. For example, we might claim that the grammar a child will learn is the one that is optimal with respect to a certain probabilistic model. We need an algorithm for computing this optimal grammar so we can check the probabilistic model's predictions and to convince ourselves we're not expecting the learner to perform magic, but we might not want to claim that humans use this algorithm. To use terminology from the grammar-based approaches mentioned earlier, a probabilistic model is a *declarative specification* of the distribution of certain variables, but it says nothing about how this distribution might actually be calculated. I think Marr's "three levels" capture this difference nicely: the question is whether we take our models to be "algorithmic level" or "computational level" descriptions of cognitive processes (Marr, 1982).

Looking into the future, I'm very excited about Bayesian approaches to language acquisition, as I think they have the potential to let us finally examine deep questions about language acquisition in a quantitative way. The Bayesian approach factors learning problems into two pieces: the likelihood and the prior. The likelihood encodes the information obtained from the data, while the prior encodes the information possessed by the learner before learning commences (Pearl, 1988). In principle the prior can encode virtually any information, including information claimed to be part of universal grammar.

Bayesian priors can incorporate the properties linguists often take to be part of universal grammar, such as X' theory. A Bayesian prior can also express soft markedness preferences as well as hard constraints. Moreover, the prior can also incorporate preferences that are not specifically linguistic, such as a preference for shorter grammars or smaller lexicons, i.e., the kinds of preferences sometimes expressed by an evaluation metric (Chomsky, 1965).

The Bayesian framework therefore provides us with a tool to quantitatively evaluate the impact of different purported linguistic universals on language acquisition. For example, we can calculate the contribution of, say, hypothetical X' theory universals on the acquisition of syntax. The Bayesian framework is flexible enough to also permit us to evaluate the contribution of the non-linguistic context on learning (Frank et al., to appear). Finally, non-parametric Bayesian methods permit us to learn models with an unbounded number features, perhaps giving us the mathematical and computational tools to understand the induction of rules and complex structure (Johnson et al., 2007).

Of course doing this requires developing actual Bayesian models of language, and this is not easy. Even though this research is still just beginning, it's clear that the details of the models have a huge impact on how well they work. It's not enough to "assume some version of X' theory"; one needs to evaluate specific proposals. Still, my hope is that being able to evaluate the contributions of specific putative universals may help us measure and understand their contributions (if any) to the learning process.

4 Conclusion

In this paper I focused on two areas of interaction between computational linguistics and linguistic theory. In the area of parsing I argued that we should design parsers so they incorporate an open-world assumption about sentences and their linguistic structures and sketched two ways in which grammar-based approaches might be modified to make them do this; both of which involve abandoning the idea that parsing is solely a process of proving the grammaticality of the input.

Then I discussed how probabilistic models are being applied in the fields of sentence processing and language acquisition. Here I believe we're at the beginning of a very fruitful period of interaction between empirical research and computational modeling, with insights and results flowing both ways.

But what does all this mean for mainstream computational linguistics? Can we expect theoretical linguistics to play a larger role in computational linguistics in the near future? If by computational linguistics we mean the NLP engineering applications that typically receive the bulk of the attention at today's Computational Linguistics conferences, I'm not so sure. While it's reasonable to expect that better scientific theories of how humans understand language will help us build better computational systems that do the same, I think we should remember that our machines can do things that no human can (e.g., count all the 5-grams in terabytes of data), and so our engineering solutions may differ considerably from the algorithms and procedures used by humans. But I think it's also reasonable to hope that the interdisciplinary work involving statistics, computational models, psycholinguistics and language acquisition that I mentioned in the paper will produce new insights into how language is acquired and used.

Acknowledgments

I'd like to thank Eugene Charniak and Antske Fokkens for stimulating discussion and helpful comments on an earlier draft. Of course all opinions expressed here are my own.

References

Steven Abney. 1997. Stochastic Attribute-Value Grammars. *Computational Linguistics*, 23(4):597–617.

A. Albright and B. Hayes. 2003. Rules vs. analogy in English past tenses: a computational/experimental study. *Cognition*, 90:118–161.

Asaf Bachrach. 2008. *Imaging Neural Correlates of Syntactic Complexity in a Naturalistic Context*. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts.

P. Boersma and B. Hayes. 2001. Empirical tests of the gradual learning algorithm. *Linguistic Inquiry*, 32(1):45–86.

Joan Bresnan. 1982. Control and complementation. In Joan Bresnan, editor, *The Mental Representation of Grammatical Relations*, pages 282–390. The MIT Press, Cambridge, Massachusetts.

Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 173–180, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *The Proceedings of the North American Chapter of the Association for Computational Linguistics*, pages 132–139.

Noam Chomsky. 1965. *Aspects of the Theory of Syntax*. The MIT Press, Cambridge, Massachusetts.

Michael Collins and Terry Koo. 2005. Discriminative reranking for natural language parsing. *Computational Linguistics*, 31(1):25–70.

Michael Collins. 2003. Head-driven statistical models for natural language parsing. *Computational Linguistics*, 29(4):589–638.

Michael C. Frank, Sharon Goldwater, Vikash Mansinghka, Tom Griffiths, and Joshua Tenenbaum. 2007. Modeling human performance on statistical word segmentation tasks. In *Proceedings of the 29th Annual Meeting of the Cognitive Science Society*.

Michael C. Frank, Noah Goodman, and Joshua Tenenbaum. to appear. Using speakers referential intentions to model early cross-situational word learning. *Psychological Science*.

Gerald Gazdar, Ewan Klein, Geoffrey Pullum, and Ivan Sag. 1985. *Generalized Phrase Structure Grammar*. Basil Blackwell, Oxford.

J. Goldsmith. 2001. Unsupervised learning of the morphology of a natural language. *Computational Linguistics*, 27:153–198.

Sharon Goldwater, Thomas L. Griffiths, and Mark Johnson. 2007. Distributional cues to word boundaries: Context is important. In David Bamman, Tatiana Magnitskaia, and Colleen Zaller, editors, *Proceedings of the 31st Annual Boston University Conference on Language Development*, pages 239–250, Somerville, MA. Cascadilla Press.

J. Goodman. 1998. *Parsing inside-out*. Ph.D. thesis, Harvard University. available from http://research.microsoft.com/˜joshuago/.

John Hale. 2006. Uncertainty about the rest of the sentence. *Cognitive Science*, 30:643–672.

Fred Jelinek. 2004. Stochastic analysis of structured language modeling. In Mark Johnson, Sanjeev P. Khudanpur, Mari Ostendorf, and Roni Rosenfeld, editors, *Mathematical Foundations of Speech and Language Processing*, pages 37–72. Springer, New York.

Mark Johnson and Eugene Charniak. 2004. A TAG-based noisy channel model of speech repairs. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 33–39.

Mark Johnson, Stuart Geman, Stephen Canon, Zhiyi Chi, and Stefan Riezler. 1999. Estimators for stochastic "unification-based" grammars. In *The Proceedings of the 37th Annual Conference of the Association for Computational Linguistics*, pages 535–541, San Francisco. Morgan Kaufmann.

Mark Johnson, Thomas L. Griffiths, and Sharon Goldwater. 2007. Adaptor Grammars: A framework for specifying compositional nonparametric Bayesian models. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 641–648. MIT Press, Cambridge, MA.

Dan Klein and Chris Manning. 2004. Corpus-based induction of syntactic structure: Models of dependency and constituency. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 478–485.

Roger Levy. 2008. Expectation-based syntactic comprehension. *Cognition*, 106:1126–1177.

John W. Lloyd. 1987. *Foundations of Logic Programming*. Springer, Berlin, 2 edition.

David M. Magerman. 1995. Statistical decision-tree models for parsing. In *The Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pages 276–283, San Francisco. The Association for Computational Linguistics, Morgan Kaufman.

Michell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

David Marr. 1982. *Vision*. W.H. Freeman and Company, New York.

Matha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Joe Pater. 2008. Gradual learning and convergence. *Linguistic Inquiry*, 30(2):334–345.

Judea Pearl. 1988. *Probabalistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, California.

Slav Petrov and Dan Klein. 2007. Improved inference for unlexicalized parsing. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 404–411, Rochester, New York, April. Association for Computational Linguistics.

Geoffrey K. Pullum. 2007. Ungrammaticality, rarity, and corpus use. *Corpus Linguistics and Linguistic Theory*, 3:33–47.

Brian Roark. 2001. Probabilistic top-down parsing and language modeling. *Computational Linguistics*, 27(2):249–276.

Stuart M. Shieber. 1986. *An Introduction to Unification-based Approaches to Grammar*. CSLI Lecture Notes Series. Chicago University Press, Chicago.

Paul Smolensky and Géraldine Legendre. 2005. *The Harmonic Mind: From Neural Computation To Optimality-Theoretic Grammar*. The MIT Press.