Linguistic **I**ssues in **L**anguage **T**echnology – **LiLT**
Submitted, October 2011

How relevant is linguistics to computational linguistics?

Mark Johnson

Published by CSLI Publications

LiLT volume 6, issue 7 October 2011

How relevant is linguistics to computational linguistics?

Mark Johnson, *Brown University*

1 Introduction

How relevant is linguistics to computational linguistics? How has the "statistical revolution" that swept computational linguistics in the 1990s affected the relationship between linguistics and computational linguistics, and how might this relationship change in the near future? These are complex questions, and this paper presents my personal perspective on them. I start by explaining what I take computational linguistics to be, and discuss the relationship between its scientific side and its engineering applications. Statistical techniques have revolutionised many scientific fields in the past two decades, including computational linguistics. I describe the evolution of my own research in statistical parsing and how that lead me away from focusing on the details of any specific linguistic theory, and to concentrate instead on discovering which types of information (i.e., features) are important for specific linguistic processes, rather than on the details of exactly how this information should be formalised. I end by describing some of the ways that ideas from computational linguistics, statistics and machine learning may have an impact on linguistics in the future.

LiLT Volume 6, Issue 7, October 2011.
How relevant is linguistics to computational linguistics?.

2 The engineering and scientific sides of computational linguistics

Before discussing the relationship between linguistics and computational linguistics, I would 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, summarisation, question-answering, etc. Like many engineering disciplines, natural language processing draws on results from a variety of scientific fields.

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). While I don't agree with this (I believe computational linguists should have a solid understanding of descriptive grammar of the kind found in e.g., Baker (1995), Huddleston and Pullum (2002) and McCawley (1988)), even if it were, would it indicate that there is something wrong with either linguistic theory or computational linguistics? I don't think it would: there's no reason to expect an engineering solution to utilise all the scientific knowledge of a related field. The fact that you can build perfectly good bridges with Newtonian physics 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 revolutionised most fields of science — but because it makes sense to think of linguistic processes as possessing an essentially computational nature. If we take computation to be the manipulation of symbols in a meaning-respecting way, then it seems reasonable to view language comprehension, production and acquisition as special kinds of computational processes, about which the theory of computation might have something important to say. 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 4 below, I think we are starting to see this happen.

3 Grammar-based and statistical parsing

In many ways the 1980s were a golden age for collaboration and cross-fertilisation 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 will 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 will describe why this happened.

The reader should be aware that I am vastly oversimplifying here by setting up grammar-based and statistical treebank parsing as two distinct categories. In reality there is a large and growing body of work on grammar-based broad-coverage parsing that bridges these two approaches, including work in LFG (Kaplan et al., 2004, Cahill et al., 2008), HPSG (Sagae et al., 2007, Cholakov et al., 2008), CCG (Hockenmaier and Steedman, 2002, Clark and Curran, 2004) and numerous variants of TAG (e.g., Shen et al. (2008)).

Some of these robust, broad-coverage parsers are perhaps more accurately described as inspired by a particular linguistic theory, rather than exactly implementing that theory. In fact it's not clear what properties a robust, broad-coverage parser needs to possess in order for it to qualify as implementing a specific linguistic theory. I am not claiming there's anything wrong with this — one can get inspiration from many places — but it does mean that one cannot use such parsers to support the claim that a particular linguistic theory supports efficient, broad-coverage parsing.

3.1 Why parse?

Rather than trying to define what it means to for a parser to implement a particular linguistic theory, I think it's more useful to clarify our goals for building such parsers in the first place. 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 the system is constructed and what counts as success.

In this paper I assume that we want to build parsers because we expect the representations they produce will be useful for various other NLP engineering tasks, such as information retrieval, machine translation, etc. This means that the parser design is itself essentially an engineering task, i.e., we want a device that returns representations of syntactic structure that are as accurate as possible for as many sentences as possible. Faithfulness to a specific linguistic theory — and the particular mechanisms that a specific theory might posit (e.g., feature-passing in HPSG, adjunction in TAG, etc.) — is usually not an important criterion in these applications. The primary concern is that the parser produces the appropriate representations accurately and efficiently.

Of course there are other reasons why one might want to build a parser. Following Stabler (1992) one might implement a specific linguistic theory as a way of checking its predictions, in which case fidelity to a specific linguistic theory is clearly of utmost importance, outweighing engineering concerns such as efficiency, robustness and broad-coverage. Similarly, if one were to build a parser as a model of on-line human sentence processing, then maximising parser accuracy might not be the primary goal; instead, the ability to predict garden-pathing and word-by-word reading times might be more valuable.

3.2 Linguistics and parsing

There are a couple of differences between the grammar-based and statistical 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 labelling and "PropBank" parsing (Palmer et al., 2005)), and, assuming one can actually reliably annotate quantifier scope, it seems reasonable to expect that 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 in-

formation 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 potential named entities) as well as the verbs that govern them, but they ignore the rest of the syntactic structure (Collins and Singer, 1999).

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, Johnson and Charniak, 2004). In this application the linguistic representations are irrelevant; instead, all that matters is how the model assigns probabilities to strings. 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 summarisation 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 approaches. (It's a different but also 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: I suspect the answer may have more to do with human factors, economics and possibly even sociology than any deep scientific principles).

Rather, I think that computational linguists working on statistical parsing may 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, and the computational linguist implementing the parsing framework doesn't actually have to understand the language-specific grammars. 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. 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.[1]

While it's true that only a small fraction of our knowledge about linguistic structure is instantiated 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. While it's true that many of the features used in statistical parsers don't correspond to linguistic constraints, it's also the case that nobody seriously claims that humans understand language only using formal syntactic linguistic constraints: human language processing is exquisitely sensitive to a wide variety of distributional and contextual information. I suspect many of the features that have been shown to be useful in statistical parsing cover psycholinguistic processing preferences (e.g., attachment preferences) and at least some aspects of world knowledge (e.g., that "apple" is likely to be the head of a direct object of "eat").

There are of course substantial differences between the aims of linguistic theory, and the aims of engineers constructing parsers. In linguistics we aim to construct a theory that *captures* appropriate generalisations about linguistic phenomena at an appropriate level of abstraction. But in statistical parsing it not necessary for the statistical model to explicitly incorporate these abstract generalisations at all: it is only necessary that the statistical features *cover* the relevant examples, possibly just by enumerating them. 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 realised 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 cover most cases of subject-verb agreement. That is, even though the Charniak-Johnson parser doesn't contain a mechanism that explicitly captures the subject-verb agreement generalisation in English, the other mechanisms it possesses permit it to cover enough of the cases of subject-verb agreement that adding an explicit subject-verb agreement mechanism did not improve overall parsing accuracy.

Note that I am 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

[1] The parsers developed by the Berkeley group are a notable exception (Petrov and Klein, 2007).

abilities (e.g., to exactly count the occurrences of millions of different configurations) 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. I think it would be extremely useful if someone developed a statistical parsing framework that made it possible to define new features and integrate them into a statistical parser without additional programming. This 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 specialised kind of inference problem. These days I view "parsing as deduction" (one of the slogans touted by the grammar-based crowd) as unnecessarily restrictive;[2] after all, psycholinguistic research shows that humans are exquisitely sensitive to distributional information, so why shouldn't we let our parsers use that kind of 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 at least the arsenal of statistical methods for parsing and learning can be applied as well to any grammatical theory.

3.3 From grammar-based to statistical parsing

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.

[2] Deduction is truth-preserving inference; probabilistic inference is strictly more general.

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 each linguistic representation to a corresponding vector of real numbers. By defining a probability distribution over such vectors we implicitly define a distribution over the corresponding linguistic representations. Abney suggested using "Maximum Entropy" distributions, which are also known as "log-linear" or "exponential" distributions, and are the same distributions used in logistic regression; this family of distributions turns out to generally fit quite well.

Importantly, the statistical procedure that learns the probability distribution over feature vectors (characterised by the feature weights $\mathbf{w}$ in Figure 1) only requires the feature vectors, and not the linguistic representations themselves. This means that as far as the probabilistic model is concerned *the details of the linguistic representations don't matter*, so long as they are rich enough to distinguish the relevant ambiguities and it is possible to compute the necessary real-valued feature 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 own work, 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 complicated programs, 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. (At the time the LFG parser I was using was unable to parse a significant fraction of the sentences in my evaluation corpus, and switching to the Charniak parser significantly improved coverage). It turns out that replacing the LFG parser in my overall system with a statistical n-best

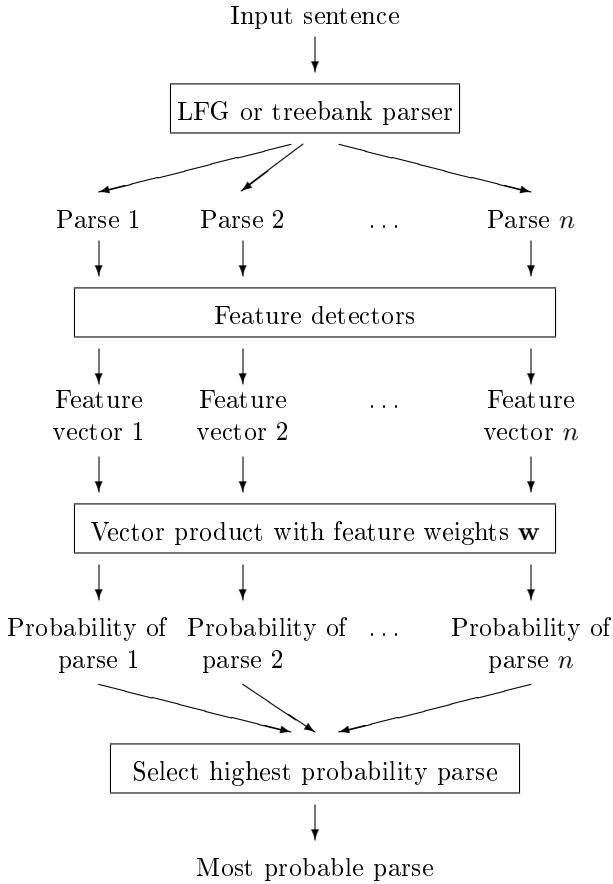

FIGURE 1 The data flow in the probabilistic LFG and treebank parsers described in the text. Notice that each parse's probability is directly determined by its feature vector and the corresponding feature weights.

treebank parser produces what Collins and Koo (2005) calls the discriminative re-ranking approach, 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.

3.4 Robust, broad-coverage grammar-based parsing

What then are the prospects for developing robust, broad-coverage grammar-based parsers? There seem to be no fundamental reasons preventing the development of such parsers, and as mentioned earlier there are a number of very impressive grammar-based parsers that do just this. The principle problem for most systems is lexical coverage (e.g., unknown words, or words requiring lexical entries not in the dictionary). This can be addressed by some kind of mechanism that generates new lexical entries on the fly, perhaps on the basis of morphological properties of the unknown word. Supertagging is widely used in such systems, both to restrict the parsing search space and possibly also to guess the syntactic properties of unknown words (Bangalore and Joshi, 1999).

I suspect these robustness and coverage problems of grammar-based parsing are symptoms of a problem in the way that parsing is usually 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.[3]

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 in-

[3] As Martin Kay pointed out to me, calling in the linguists often makes matters worse. For example, a linguist might notice the count/mass distinction in noun phrases, and suggest that the parser's categories be refined to express this. While one might hope that constraints associated with this distinction will disambiguate some sentences, the usual effect of such refinements, however, is to introduce a further ambiguity into the grammar, and significantly increase the average number of parses per sentence.

tended 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. It's not useful for a parser to simply state that the sentence is ungrammatical. Note that the probabilistic models employed by statistical parsers provide a way of comparing a pair of parses (in terms of their probabilities), but since every possible parse receives non-zero probability (because of smoothing) they don't rule out any structure absolutely.

The grammars used in grammar-based approaches typically distinguish grammatical from ungrammatical analyses by explicitly characterising 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 ahead of time 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, that shouldn't stop it from interpreting the rest of the sentence. Statistical parsers are considerably more open-world in this respect. 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 grammaticality-based grammars used in grammar-based parsing mean we have to abandon such grammars? 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.

The first approach, which is already implicit in some parsers, is *grammar relaxation*. We can obtain an open world model by relaxing our

interpretation of some or all of the constraints in the grammar. Instead of viewing grammatical constraints as hard constraints that define the set of grammatical constructions, we reinterpret them as violable features of analyses, and perhaps associate a cost with them. 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 penalises analyses in which subject-verb agreement fails. Instead of assuming that each verb comes with a fixed set of subcategorisation requirements, we might view subcategorisation as preferences for certain kinds of complements, implemented by features in an Abney-style statistical model. Unknown words come with no subcategorisation 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. "Unknown word guessers" and "fragment parsing mechanisms" might be viewed as performing grammar relaxation; the grammar only contains a certain set of rules and lexical entries, but if they don't suffice to parse a particular sentence, these devices "cons up" new rules and lexical entries to do the job.[4] While the unknown word guessers and fragment parsers used today are typically relatively ad hoc, we may be able to develop a more principled approach by incorporating ideas from the statistical models of the acquisition of the lexicon and the grammar discussed in section 4 below.

While grammar relaxation is relatively standard and familiar, it may be worth considering other approaches as well. One disadvantage of the grammar relaxation approach is that it effectively abandons the notion of grammaticality. One way to achieve an open-world approach to parsing while maintaining the standard closed-world conception that grammars generate only grammatical analyses is to abandon the claim that a parse is a proof of the grammaticality of the input sentence. One way to do this is to *incorporate explicit models of disfluencies into the parsing process*. We can use a *noisy channel* to map grammatical analyses generated by the grammar to the actual input sentences we are given to parse. Mathematically, one defines a generative model $\mathrm{P}(G)$ that generates underlying grammatical analyses G, which are then passed to a disfluency model $\mathrm{P}(D|G)$ that can introduce disfluencies, producing

[4] The smoothing techniques standardly used to estimate the models used by statistical parsers might also be seen as relaxation: we don't have to assign an event zero probability just because it was not observed in the training corpus.

potentially disfluent analyses D.

$$\mathrm{P}(G, D) \quad = \quad \mathrm{P}(G)\,\mathrm{P}(D \mid G)$$

Parsing involves inverting this generative process to recover the underlying grammatical source sentence as well as its structure, as this would permit us to semantically interpret the potentially disfluent sentence. Given a sequence of input words w, it seems reasonable to find both the most likely disfluent analysis $d^\star(w)$ and corresponding underlying grammatical analysis $g^\star(w)$ of w (other objectives might involve marginalising over one of these structures). By Bayes rule, these are given by:

$$(g^\star(w), d^\star(w)) \quad = \quad \mathrm{argmax}_{(g,d)\,:\,W(d)=w} \mathrm{P}(G = g)\;\mathrm{P}(D = d \mid G = g),$$

where $W(d)$ is the yield or string of words associated with disfluency analysis d. Such an approach can regarded as formalising the idea that ungrammatical sentences are interpreted by analogy with grammatical ones.

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 model 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. Because the system optimises the joint probability of the disfluency and the underlying grammatical analysis, it is possible that the system could choose a disfluency analysis of a grammatical sentence. As Steven Abney (p.c.) points out, this might be reasonable in some circumstances: "thanks for all you help" is grammatical, but it might be preferable to interpret it as an erroneous rendition of "thanks for all your help".

Johnson and Charniak (2004) describe a noisy channel model for interpreting transcribed speech that functioned along the lines described above. In that system the disfluency model introduces *restarts* (e.g., "You get, uh, you can get a car ...") and *repairs* (e.g., "I want a ticket to Boston, uh, to Denver on Friday") (Shriberg, 1994) and used a version of the Charniak parser trained on the Switchboard corpus with disfluencies excised as a model of grammatical analyses. Johnson et al. (2004) explain how to generalise this model to make it sensitive to the syntactic locations of the disfluencies.

Computationally, I suspect that moving from a closed-world to an open-world approach to parsing will require a major rethinking of the

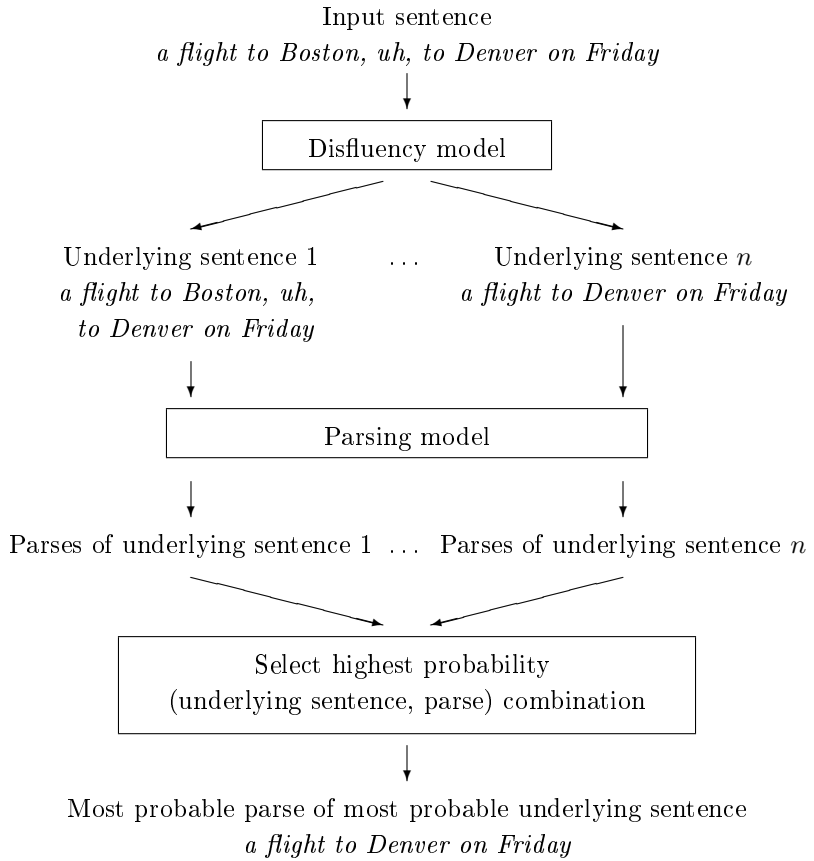

FIGURE 2 The data flow in the noisy channel model for parsing input with speech disfluencies described in Johnson and Charniak (2004). In the version of the model described in Johnson et al. (2004) the disfluency model is sensitive to the syntactic location of the disfluency (i.e., the disfluency model operates on trees rather than strings).

parsing process. Notice that all of these 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 search) will be useful; indeed, these methods are being used in state-of-the-art CCG, LFG, HPSG and TAG systems today.

4 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 of these 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 (see Headden III et al. (2009) for recent work in this area), 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 will 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 generalises is just as important as specifying the learning algorithm itself. One of the things I like about this work is that it may help us get beyond the naive nature-versus-nurture arguments that characterise 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 modelling inter-word dependencies improves word segmentation, while Johnson (2008b) investigates the role that synergies between such dependencies and other kinds of linguistic structure (such as syllable structure) might play in acquisition. One of the exciting things about this work is that it permits a quantitative evaluation of the contribution that specific linguistic representations or constraints might make to the learning process; it will be very interesting to see if we can demonstrate a crucial role for particular linguistic universals.

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. Currently it seems very hard to characterise the non-linguistic contextual information available to a child learning a language, let alone specify precisely how it might be used in the acquisition process. However, it does seem reasonable to expect that it plays a less crucial role in, say, the acquisition of phonology than it does in the acquisition of syntax.

Further, 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 optimisation into linguistic theory in a way that linguists could understand, and this serves as a very important bridge for them to understanding probabilistic models. There is a very close mathematical connection between Smolensky's "Harmony Theory" and Abney's probabilistic models, with Optimality-theory "constraints" corresponding to statistical "features" (Goldwater and Johnson, 2003).

As I argued above, it's important with any computational modelling 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 learn 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 about the innate information available to the child and the kinds of information in the input that the child is sensitive to. Because these probabilistic models are too complex to analyse analytically in general we need a computational procedure to compute the model's predictions, but the computational procedure itself need not be 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.[5] 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).[6]

Looking into the future, I am 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 innate information claimed to be part of universal grammar.

[5] This is reminiscent of *ideal observer analysis* in psychology, although in this case we are modelling the child as ideal learner that extracts all of the information present in his or her input.

[6] In my opinion Marr's "computational level" is poorly named; the "computational level" is concerned with representations, or more precisely, the information that the representations contain, not the computations that manipulate these representations. I prefer to call Marr's upper level the "informational level".

Bayesian priors can incorporate the properties linguists often take to be part of universal grammar, such as X' theory and constraints on movement. 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., 2009, Jones 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, Johnson, 2008a, Johnson and Goldwater, 2009).

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 formulate specific proposals to evaluate. Still, my hope is that being able to evaluate the contributions of specific putative universals (linguistic or otherwise) may help us measure and understand their contributions (if any) to the learning process. I expect empirical investigation will show learning process to be far more complex — and interesting — than the current "nature versus nurture" debate suggests.

5 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 modelling, with insights and re-

sults 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 am 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, language acquisition and linguistic theory described in this paper will produce new insights into how language is acquired and used.

Acknowledgments

I would like to thank Stephen Abney, Eugene Charniak, Katherine Demuth, Antske Fokkens, John Maxwell, Yusuke Miyao, Mark Steedman and the LiLT reviewers for their stimulating discussion of this topic and thoughtful comments on earlier drafts of this paper. Of course all opinions expressed here are my own. This material is based upon work supported by National Science Foundation under Grant Numbers 0544127 and 0631667.

References

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

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

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

Baker, C.L. 1995. *English Syntax*. Cambridge, Massachusetts: The MIT Press, 2nd edn.

Bangalore, Srinivas and Aravind K. Joshi. 1999. Supertagging: An approach to almost parsing. *Computational Linguistics* 25(2):237–265.

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

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

Cahill, Aoife, John T. Maxwell III, Paul Meurer, Christian Rohrer, and Victoria Rosén. 2008. Speeding up LFG parsing using c-structure pruning. In *Coling 2008: Proceedings of the workshop on Grammar Engineering Across Frameworks*, pages 33–40. Manchester, England: Coling 2008 Organizing Committee.

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

Charniak, Eugene 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: Association for Computational Linguistics.

Cholakov, Kostadin, Valia Kordoni, and Yi Zhang. 2008. Towards domain-independent deep linguistic processing: Ensuring portability and re-usability of lexicalised grammars. In *Coling 2008: Proceedings of the workshop on Grammar Engineering Across Frameworks*, pages 57–64. Manchester, England: Coling 2008 Organizing Committee.

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

Clark, Stephen and James R. Curran. 2004. Parsing the WSJ using CCG and log-linear models. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 103–110. Barcelona, Spain.

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

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

Collins, Michael and Yorav Singer. 1999. Unsupervised models for named entity classification. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP 99)*.

Frank, Michael C., 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*.

Frank, Michael C., Noah Goodman, and Joshua Tenenbaum. 2009. Using speakers' referential intentions to model early cross-situational word learning. *Psychological Science* 20:579–585.

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

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

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

Goldwater, Sharon and Mark Johnson. 2003. Learning ot constraint rankings using a maximum entropy model. In J. Spenader, A. Eriksson, and O. Dahl, eds., *Proceedings of the Stockholm Workshop on Variation within Optimality Theory*, pages 111–120. Stockholm: Stockholm University.

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

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

Headden III, William P., Mark Johnson, and David McClosky. 2009. Improving unsupervised dependency parsing with richer contexts and smoothing. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 101–109. Boulder, Colorado: Association for Computational Linguistics.

Hockenmaier, Julia and Mark Steedman. 2002. Generative models for statistical parsing with combinatory categorial grammar. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 335–342. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics.

Huddleston, Rodney and Geoffrey K. Pullum. 2002. *The Cambridge Grammar of the English Language*. Cambridge, UK: Cambridge University Press.

Jelinek, Fred. 2004. Stochastic analysis of structured language modeling. In M. Johnson, S. P. Khudanpur, M. Ostendorf, and R. Rosenfeld, eds., *Mathematical Foundations of Speech and Language Processing*, pages 37–72. New York: Springer.

Johnson, Mark. 2008a. Unsupervised word segmentation for Sesotho using adaptor grammars. In *Proceedings of the Tenth Meeting of ACL Special Interest Group on Computational Morphology and Phonology*, pages 20–27. Columbus, Ohio: Association for Computational Linguistics.

Johnson, Mark. 2008b. Using adaptor grammars to identifying synergies in the unsupervised acquisition of linguistic structure. In *Proceedings of the 46th Annual Meeting of the Association of Computational Linguistics*. Columbus, Ohio: Association for Computational Linguistics.

Johnson, Mark 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.

Johnson, Mark, Eugene Charniak, and Matthew Lease. 2004. An improved model for recognizing disfluencies in conversational speech. In *Rich Transcription 2004 Fall workshop (RT-04F)*. Palisades, NY.

Johnson, Mark, 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.

Johnson, Mark and Sharon Goldwater. 2009. Improving nonparameteric Bayesian inference: experiments on unsupervised word segmentation with adaptor grammars. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 317–325. Boulder, Colorado: Association for Computational Linguistics.

Johnson, Mark, 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, eds., *Advances in Neural Information Processing Systems 19*, pages 641–648. Cambridge, MA: MIT Press.

Jones, Bevan, Mark Johnson, and Michael Frank. to appear. Learning words and their meanings from unsegmented child-directed speech. In *The Proceedings of the 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*.

Kaplan, Ron, Stefan Riezler, Tracy H King, John T Maxwell III, Alex Vasserman, and Richard Crouch. 2004. Speed and accuracy in shallow and deep stochastic parsing. In D. M. Susan Dumais and S. Roukos, eds., *HLT-NAACL 2004: Main Proceedings*, pages 97–104. Boston, Massachusetts, USA: Association for Computational Linguistics.

Klein, Dan 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.

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

Lloyd, John W. 1987. *Foundations of Logic Programming*. Berlin: Springer, 2nd edn.

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

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

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

McCawley, James D. 1988. *The Syntactic Phenomena of English*, vol. volumes 1 and 2. Chicago: The University of Chicago Press.

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

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

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

Petrov, Slav 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: Association for Computational Linguistics.

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

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

Sagae, Kenji, Yusuke Miyao, and Jun'ichi Tsujii. 2007. Hpsg parsing with shallow dependency constraints. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Prague, Czech Republic: Association for Computational Linguistics.

Shen, Libin, Lucas Champollion, and Aravind K. Joshi. 2008. LTAG-spinal and the Treebank: A new resource for incremental, dependency and semantic parsing. *Language Resources and Evaluation* 42:1–19.

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

Shriberg, Elizabeth. 1994. *Preliminaries to a Theory of Speech Disfluencies*. Ph.D. thesis, University of California, Berkeley.

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

Stabler, Edward P. 1992. *The Logical Approach to Syntax: Foundations, specifications and implementations*. Cambridge, Massachusetts: The MIT Press.