

# Building lamp-posts\*

Mark Johnson  
Brown University

Australian Language Technology Association meeting  
10th December 2003

The organizers asked me to talk about the impact that I think computational linguistics will have on neighboring disciplines and on its broader intellectual implications for society in general. I guess it's no surprise that I'm going to argue that computational linguistics is relevant in a wide range of ways: after all, the organizers are not likely to have invited me to address you tonight unless they thought I would have something up-beat to say.

So while I would agree that we will all live better, if not longer, thanks to computational linguistics (especially if the research funding agencies keep up their financial support), my real message to you tonight is computational linguistics is an exciting and incredibly diverse field to be involved in, especially right now, and there's good reason to believe that some surprising results are just around the corner. The diversity of the field is something I will especially emphasize: a new breakthrough can both lead to the founding of a start-up company and also tell us something about philosophical questions first posed by the ancient Greeks.

But let me begin by telling you a bit about the field. The field has two sides to it: an academic side, and a more applied side. The academic side is called "computational linguistics" and studies the computational processes involved in language comprehension, production and learning. The applied side is called "natural language processing", although to be honest I suspect that the two terms are used pretty much interchangeably. Anyway, the applied side of the field investigates how computers can do useful processing of text and speech, such as *machine translation* (the automatic translation of text or speech from one language to another by computer), *information extraction* (where a computer extracts information from texts to put into a database) and *question-answering* (where a computer formulates answers to natural language questions on the basis of information in its databases).

While I haven't conducted a poll, my guess is that most of the people in the field are there because they find it fun to make computers do cool things with language, such as building a machine that can listen and talk back. This is the applied side of the field, and it is growing into a commercially important industry. But I suspect that a good fraction of us are doing computational linguistics because we believe that there is something unique and important about the combination of computation and language that is likely to lead to new insights about both language and the mind. The very name "computational linguistics" indicates

---

\*I'd like to thank Steven Bird of Melbourne University and Eugene Charniak of Brown University for their generous suggestions, most of which were included below. All responsibility for errors remains my own.

the field's mixed heritage of computer science and linguistics: it is a hybrid of a hard science and the humanities that is unique as far as I know.

Why does it makes sense to mix computers and linguistics? Is it just that we use computers a lot that justifies the “computational” in “computational linguistics”.

I don't think so; after all, it's hard to imagine a scientific field today where computers don't play a central role. Rather, I think it is because there is something essentially *computational* about the processes involved in language itself. Language is a tool – I would argue by far the richest and most flexible tool – that humans have for conveying information from one to another. And this process of conveying information – of a thought in one person leading to an utterance or a text leading to a corresponding thought in another person – is, as far as I can tell, essentially *computational* in nature, since the essence of computation is the manipulation of information. So anything whose primary purpose is the manipulation or transformation of information – language is a prime example, but genomics is another, with close ties to computational linguistics, it turns out – has a deeply computational aspect to it.

It is quite possible, it seems to me, that the kinds of computation involved in language understanding and production performed by the brain are completely unlike those we currently know of and have built into our computers. But there's nothing wrong with that – after all, that only means that there are richer and more complex kinds of computation to explore.

Indeed, I think that if one wants to explore the complexity of the human mind, it makes sense to start by studying human language. In a certain sense, language is a pure product of the mind, richer and less externally constrained than any other psychological process. Its structure must come from the mind somehow, and it's not unreasonable to suppose that linguistic structure reflects the mental structure of thoughts in some not yet fully understood way.

The field of computational linguistics was born shortly after the invention of computers in the 1950s. At that time there was a tremendous optimism that the war-time breakthroughs in cryptography and computers would solve all sorts of complicated problems. For example, it was hoped that code-breaking algorithms could translate Russian into English by viewing Russian as being just being some very tricky code for their English translations.

I wasn't around when all of this was happening, but it seems as if there were two major intellectual streams active at the time that are still driving the field today.

First, Turing and von Neumann had come up with the idea of the computer. It's hard for us to imagine what a break-through this was: until that point machines were things that manipulated other *things*, while a computer is a machine that manipulates pure *information*. In the same way physicists studying thermodynamics developed the theory of the “heat engine” as an idealization of real engine, these early computer scientists developed the theory of automata as idealized models of actual computing machines, and it turned out that abstract or formal languages were the natural products of these idealized machines. So the connection between computers and language goes way back to the birth of computers itself.

The second major intellectual advance at the middle of the last century was *information theory*, developed by Shannon and colleagues, which explained how the statistical properties of messages affect how they can be transmitted over an information channel. One of their key

insights is that one could devise an optimal way of recovering a corrupted or distorted signal using statistical methods; this *noisy channel model* plays an important role in computational linguistics today.

Even in the 1950s there was a certain tension between these lines of work, and it is this tension that I will be focusing on in the rest of this talk. Information as conceived of in the computational setting by Turing and von Neumann required it to be digital and discrete, while Shannon’s notion of information focused on its role in communication, emphasized its broad statistical properties, and was equally compatible with both continuous analog signals as well as digital ones.

Noam Chomsky became a dominant figure in the field in the early 1960s, and his effect on linguistics and computational linguistics is hard to over-estimate. He emphasized the connection between automata, languages and grammars, and his technical work on generative grammar still drives a large part of the field of linguistics today. Getting back to the topic, Chomsky suggested that it was highly unlikely purely statistical approaches would lead to deep insights into the way that language works. For example, Chomsky pointed out, in most people’s experience, the two sequences of words “I want a fragile green elephant” and its reverse “elephant green fragile a want I” have exactly the same frequency of occurrence, namely zero – neither has ever been heard before – but the first is obviously a sentence of English, but the second one is not; to use the technical term, it is just word salad. Chomsky also emphasized to the creative aspects of language use – the way that we can effortlessly assemble words to form novel sentences that have never been uttered before. At the time it seemed that this sort of unbounded creativity was beyond the capability of any statistical model, and this was taken as a powerful argument for Chomsky’s own non-statistical generative grammar.

Chomsky also drew attention to the importance of language acquisition. By the age of three or four the vast majority of human children wind up speaking a language extremely well; something which remains beyond the grasp of chimps and other animals even after years of intensive instruction. Chomsky’s question was: how do kids do this? What exactly does a child have to learn when they are learning a language like Chinese or English?

His answer was (and still is): only the language-particular details of Chinese or English or whatever, anything which is common or *universal* to all languages probably doesn’t have to be learned, but is hard-wired in to each and every one of us, presumably encoded somehow in the genome as an instinct of some kind, much in the way that, e.g., the bees’ dance is an instinct inherited by bees. Chomsky imagined that language learning is a matter of merely learning the vocabulary of the language concerned (although this alone is a major task), plus setting the value of a few binary switch-like parameters, which encode linguistic facts like “does the verb come last or not in my language” (“last” is the answer a Japanese-speaking child would have to learn, “not last” is what an English speaker would learn).

Chomsky’s approach is one known to the philosophers as “rationalism”, and which goes back in one form or another to Plato and before. The rationalists’ idea is that knowledge – such as our unconscious knowledge of a language or whatever – is not learned from experience, but is innate or born into us.

The opposing point of view is known as *empiricism*, which holds that our knowledge comes from our experience of the world; our mind is a *tabula rasa* or blank slate, and the

rich structure of our languages and our minds reflects the rich structure of the environment in which we are embedded.

To most of the intellectual world, including most linguists and even computational linguists, rationalism and empiricism are viewed as two diametrically opposed philosophies, with the rationalists arguing that knowledge, especially linguistic knowledge, is innate, while the empiricists argue that it is learned by some kind of suitable procedure.

Usually statistical and probabilistic approaches are conflated with empiricism, as one of the obvious uses of statistical methods is to extract reliable generalizations from a set of observations. Rationalism seems to have no need for probabilities, as it assumes that general learning procedures of the kind that statistics offers contribute little to our knowledge of language.

But notice that this conflation is not necessary: one can easily imagine probabilistic knowledge being innate. The Bayesian statistical framework in particular is quite compatible with innate knowledge being probabilistic in nature. In this framework learning involves the interaction of a set of empirical observations and a prior distribution, which can reflect innate probabilistic biases or tendencies to make certain types of generalizations from this data rather than others.

Similarly, rationalist approaches usually assume more complex and intricate representations for language, and empiricist approaches usually adopt simpler representations. Again, the reason is not hard to see: the empiricists have to posit mechanisms that can learn whatever representations they assume, so they have a powerful motivation to keep their representations simple. But for the rationalists the representations come for free (or at least until we learn more about the constraints on genetically encoded knowledge), so they have felt less restraint in positing arcane mental representations, and their theories of linguistics are correspondingly more baroque.

In the 1980s rationalism reigned supreme in linguistics and related fields, and computational linguistics was largely concerned with building computer programs that could compute the complex representations the rationalist linguists were positing. Typical studies in linguistics and computational linguistics looked at just a handful of constructions in great depth usually selected for their linguistic interest rather than, say, their frequency or importance in some kind of application. Indeed, the argument was often made that the scientifically most interesting constructions may well be extremely rare (just as in, say, biology, where extremely rare plants are often of great scientific importance).

The general rationalist-inspired view at the time was that language and language processing should be regarded as a kind of specialized logical system. The innate knowledge of language specify the logic, the grammar and parameter settings correspond to axioms, and language processing (i.e., comprehension, production or acquisition) corresponds to deduction or inference in this specialized logic.

The cracks in the rationalists reign in computational linguistics started to appear in the 1980s from two major directions.

First, a new type of computational model known as *neural networks* started to gain popularity in cognitive science. (Neural networks are so-called because they were inspired by the kinds of networks that neurons form in the brain, but most neural networks posited in computational linguistics make no claim to biological plausibility). While very restricted

in the types of representations they can compute – they can’t describe the diversity or creativity of our use of language that Chomsky drew attention to – neural networks had one big attraction relative to rationalist approaches: they could learn from experience. Neural networks showed that empiricism ought to be taken seriously again, even if there are serious gaps in neural networks’ capabilities.

What is it that enables neural networks to learn? The crucial difference is that neural networks used soft, continuously varying quantities in their calculations, while the rationalist grammatical models used boolean values that are either off or on. Learning in a neural network proceeds in a gradual fashion, slowly adjusting the quantities in order to make the network perform slightly better after each experience. This *incremental learning* is not possible with Chomsky’s generative grammars: the switch-like parameters have to be either off or on, not somewhere in between.

The second and more important development (for computational linguistics at least) occurred in the closely related field of speech recognition. Just as in computational linguistics, the early work in speech recognition was in the rationalist tradition. But in the 1980s researchers at IBM and elsewhere began experimenting with a certain kind of statistical framework known as *hidden Markov models*, which are not unlike the statistical models criticized earlier by Chomsky because all they did was track dependencies between adjacent words and ignored the global structure of sentence. And yet these models produced speech recognizers that worked much better than the rationalist approaches that went before them.

I think that there were three crucial aspects to the success of these statistical models. First, the models used continuously variable parameters just like neural nets, rather than binary parameters, which meant that they could be adjusted or tuned to optimize performance; a simple but very important kind of learning. Second, the models were tuned to perform best on the constructions occurring most frequently in the speech that the system would actually be used to recognize – remember that the rationalists tend to focus on esoteric and unusual constructions – so it’s not surprising that average performance improved. Third, the system’s performance was actually measured quantitatively, so it was possible to evaluate whether each new change actually improves overall performance; in other words, the speech researchers adopted a harsh, thoroughly empirical approach to the task of developing the best speech recognizers possible.

Largely inspired by the successes of the empirical approach in speech recognition, an empiricist revolution swept through computational linguistics in the mid-1990s. It reinvigorated research into linguistic corpora. A *corpus* is just a large collection of text or speech: the ones I work with most are a collection of about a year’s worth of newspaper articles and transcripts of telephone conversations. Corpora are used both to provide the data for training statistical learning procedures and also for quantitative evaluation of a system’s performance, i.e., measuring just how well it is performing. These days a small corpus might contain a million words of text, but modern speech recognizers are likely to be trained from several billion words of text, for reasons I’ll get to shortly. This work made people recognize just how much interesting information can be extracted using relatively simple, superficial methods from large amounts of text.

For example, consider the task known as *word sense disambiguation*, which is identifying which meaning of an ambiguous word like “bank” was intended in a particular context. In

principle this could be a very hard problem: one can easily construct examples in which disambiguation requires full, deep comprehension of the whole context. However, it turns out that most of the time the other words in the context provide enough information to disambiguate: if “bank” is being used in the river sense it is usually surrounded by words like “sand” or “grassy”, while if it is being used in the financial sense it is usually surrounded by finance-related words like “account”. And one can collect these words straight from appropriate corpora: the words you find in the Financial Times are likely to be good examples of finance-related words.

This renewed focus on empirical phenomena turned out to be extremely productive. For many practical tasks statistical methods turn out to be just what the doctor ordered. Tasks like document retrieval, which is retrieving a set of documents relevant to a particular query – what a Web search engine does – can be performed fairly well using relatively simple statistical methods.

While at first glance one might imagine that deep linguistic analysis of the query and the documents might be useful, this turns out not to be the case. Most Web queries are extremely short – I remember hearing from someone working in a search engine company in the 1990s that 40% of the search queries were just single words (and 20% were the single word “sex”), and there’s just not that much deep linguistic analysis that one can do on a single isolated word. Retrieval also does not benefit from deep linguistic analysis of the documents being retrieved. Our understanding of semantics – the meaning of a text – just isn’t good enough for us to be able to do much better than characterizing the meaning of an arbitrary text by the bag of words that occur in it. So for many practically important tasks like document retrieval, relatively simple statistical methods may offer close to optimal performance for the foreseeable future.

By the late 1990s empiricism seemed not only to be resurgent, but also dominated the field. Today one still hears talk about rationalism versus empiricism, and more often than not the conclusion is that empiricism won.

But I think that is missing the bigger picture. Instead, I think the real action is in novel ways of combining ideas from both the empiricist and rationalist approaches that lead to methods that perform better than either approach alone and, who knows, just might lead to new ways of looking at the millenia-old rationalist-empiricist debate.

In the early 1990s Mitch Marcus and colleagues at the University of Pennsylvania assembled a new type of corpus not just of words but of complete linguistic analyses: analyses with the kind of details posited by Chomskyian grammarians, who are solidly in the rationalists’ camp. This made it possible to investigate whether the empiricists’ statistical techniques could be used to learn the kinds of detailed linguistic representations that the rationalists had argued for. Notice that in a sense this work is breaking some of the usual assumptions about the rationalist-empiricist divide by training empiricist methods on the complex representations usually associated with rationalist approaches. Well, it turns out that by using the right sorts of statistical models and the right kinds of corpora to train the models on, that yes, it is possible for a statistical parser to produce analyses of the kind that the rationalists had argued for, and that the resulting hybrid models produce results that are better – more accurate – than purely rationalist or empiricist models alone.

One interesting turn is that these new statistical parsers are being applied to improve

the accuracy of speech recognition. In fact, there do seem to be limits to what purely empiricist methods can achieve. The *curse of dimensionality* refers to the fact that the amount of data required to train a model grows exponentially with its complexity. Consider those speech recognition models I spoke of earlier, which function by collecting statistics on adjacent tuples of words. Suppose we want to build a speech recognizer with a 1,000 word vocabulary. If the model tracks all possible pairs of words it will need to collect statistics on  $1,000 \times 1,000$  or 1 million different word pairs, a relatively easy task with today's corpora and computers. If we want to make our model more accurate by tracking all word triples then we will need to track  $1,000 \times 1,000 \times 1,000$  or 1 billion word triples, which is starting to become a non-trivial task for which we one needs the multi-billion word corpora I mentioned earlier. You can probably see where this is going: improving the model by incorporating four word combinations involves tracking one trillion quadruples, and probably no one has the resources to build a model that tracks all possible quintuples of words.

It's interesting that the statistical parsers I mentioned earlier – the ones trained on the corpus of rationalist linguistic analyses – seem to offer a way out of this dilemma. It's not the case that all one billion possible word quadruples are equally important, but a simple statistical model that just counts tuples of words has no way of knowing in advance which interactions are important and which ones aren't. Well, it turns out that the rationalist representations do identify which words are related and how they are related, so they can focus their attention, so to speak, on just the few important interactions, and by identifying a smaller number of important interactions to track they side-step the curse of dimensionality, so to speak. And in the research lab at least, speech recognizers incorporating these rationalist-inspired but empiricist-learned models outperform comparable systems that don't identify and selectively focus on these important interactions. Personally I find this a particularly nice turn of events in which computational linguistics is giving something back to speech recognition, in return for the important contribution – the empiricist approach – that speech recognition made to computational linguistics.

And while, as they say, it is hard to predict, especially the future – my guess is that new combinations of empiricism and rationalism are going to be even more important in the future. Recall the document retrieval scenario I mentioned earlier, in which deeper linguistic analysis seems to offer no foreseeable benefits. Well, think of what will happen in the near future when computers shrink to the size of cell phones or fountain pens, without the space for either a keyboard or a display screen. The obvious way to communicate with such small computers is by speech. Spoken queries tend to be longer than typewritten ones – they will be phrases or full sentences – and so can benefit from deeper linguistic analysis (we've already seen that this improves speech recognition accuracy). The same sort of thing holds for the material the retrieval program should return. While simply returning a whole document as an answer to a query is fine when you've got plenty of screen real-estate to display it, it's not reasonable when the machine is going to be reading the answer out to you. For example, if I asked the machine “when was John Howard first elected to parliament?” I want a date as an answer, not his entire biography. And deeper linguistic analysis does seem to be useful for identifying just which phrase in a document is the answer to a specific question.

So I think it is reasonable to assume that this new mix of empiricist and rationalist approaches will be extremely productive; *machine translation* is another area in which I

expect that this combination will pay off.

But what does this have to say about the age-old rationalist versus empiricist debate? Here I am not so sure what the answer will be. I'm fairly certain that the recent work in computational linguistics I just mentioned shows that there is a much larger number of possible ways in which linguistic and mental processes might come about than previously imagined. Specifically, I think it's fairly clear that statistical inference or some other kind of reasoning that can take advantage of soft, violable constraints, is extremely useful when dealing with noisy or incomplete information – and it turns out that much of the time the information available to us is incomplete or uncertain. It's interesting that psycho-linguists have experimental evidence that humans too make use of statistical tendencies as well as hard rules during language comprehension.

Ultimately though I would have to admit that nothing I know of so far fundamentally contradicts the rationalist approach. The empiricist learning techniques we know of that work so well are all basically just methods for estimating the values of continuous parameters. The thing that was surprising about the empiricist revolution, to me at least, is how many different processes and tasks can be formulated as parameter estimation tasks.

But so far at least these methods don't seem to be able to learn from scratch the complex representations of the rationalists; estimating parameters is one thing, but learning the appropriate complex structures from scratch seems to be something beyond their capabilities. Instead, these empiricist methods need a corpus of these rich representations from which to learn, and nobody provides human children with anything like such a corpus when they are learning language.

Of course, it is quite possible that someone – perhaps someone in this audience – will discover a new and more powerful learning method that is capable of inferring structure as well as estimating parameters. After all, we've only been investigating this mixture of empiricism and rationalism for a decade so far.

So I am extremely optimistic about the new mixture of empiricism and rationalism in computational linguistics. I think we've only begun to scratch the surface of its possibilities. We have found a way to bring the empiricists' box of learning and empirical evaluation tools to bear on questions posed by the rationalists by building specialized corpora. Each corpus focuses on a set of issues and illuminates a surrounding set of questions much like the way a lamp-post casts light on an otherwise dark landscape. These corpora are still sufficiently rare and expensive that there's undoubtedly a fair amount of "looking under the lamp-post" going on; in other words, investigating a particular question just because an existing corpus can answer it. It's also true that after a corpus has been completed we sometimes discover that it's not optimally designed to answer all of the questions we want to ask. But even though I admit that the critics sometimes have a point, I also think that they miss the point. What's really important about this new work is that we're learning, for the first time ever, how to build lamp-posts.