Computational linguistics: what about the linguistics?

Karen Spärck Jones
Computer Laboratory, University of Cambridge
William Gates Building, JJ Thomson Avenue, Cambridge CB3 0FD, UK

This paper in its final form will appear in *Computational Linguistics*, 2007.

Linguistics and computation

Three times a year I get my copy of a wholly respectable mainstream linguistics journal. Its scholarly articles are rich in examples from varied languages, and alongside these detailed analyses it advances theoretical claims and counterclaims. Its many reviews point to much more of the same.

But this journal content is of interest here for another reason than these scholarly ones. First, references to computing are conspicuous by their absence. Just occasionally, grammar types or semantic models appear that have computational connections, for example in a shared view of feature sets; and there are references to computational corpus analysis, though more often in reviews than in major articles. Very very occasionally, there are papers that are more manifestly computational ones, for instance in applying unification to syntactic structures. But in general, the notion that computation in a serious sense, not just as some highly abstract grounding or, maybe, politically correct meta-reference, has something important to say to linguistics never figures.

Does this matter? Specifically, whether it matters to the linguists or not, does it matter to us (computational types)? In a world of proliferating specialist journals and, increasingly, conference proceedings, why worry about the lack of computational reference in mainstream linguistics journals? Being computational is not the only way of being legitimate in any field, but there are plenty of publishing venues for the computational. So it need not be a criticism of a linguistics journal like the one I get that it does not have more on computational linguistics, or natural language processing, or natural language engineering, or human language technology. We may find the lack of computational reference a surprise, but even if there are a lot of non-computational linguists out there (as there indubitably are), why should this be a problem?

The linguists may argue they have things we ought to learn from them, and it's up to us to bridge the gap. We may argue just the reverse. But in my view there is a deeper problem. This has to do with the fact that as far as I can tell, the journal I have referred to, and other linguistics journals, are dominated by some Chomskyan paradigm. But does it matter that the linguistic zeitgeist is Chomskyan? Further, does it matter precisely which Chomskyan paradigm - older or newer, broader or more specific, pure or modified?

Why should this zeigeist matter to us? Whether or not the linguists could learn from us, does this pervasive Chomskyan flavour stop us learning beneficially from them? In particular, does it stop us learning not just from the detailed data they may present, but from their theories and models, giving us necessary underpinnings for our applications?

There are two reasons for saying it shouldn't matter, one bad, and one good.

The bad one is that there isn't actually a problem: au fond, we are all Chomskyans now, at least as dealers in language and setting aside cognition. That is to say we all believe in *some* sort of deep and surface layers and some sort of formal rule that connects some sort of symbolic object in the one with an object in the other. But what we all share, put thus, makes such a weak generalisation it doesn't do away with the problem.

The good reason is this: the crucial problem is that computational linguistics is fundamentally, and essentially, about *process*, about working with language, and linguistics as she is spoke is not about language processing. Just as in computation in general, process is not the mere implementation of some non-process account of something, but supplies the motivational account. Thus in computational linguistics, the mechanisms for selecting word senses provide interpretations for what it is for a word to have a sense in relation to a linguistic structure, and for what a linguistic structure is when built by disambiguation. Linguistics in its mainstream forms is not about process, i.e. about algorithmic process: computational linguistics is about process, and in a more thorough sense not just than Chomskyan performance, but than "computation" as a generic abstraction.

Martin Kay is reported to have said, in his Lifetime Award speech in 2005, that computational linguistics, as opposed to natural language processing, is about using computing to advance linguistic theory. The question is whether using computing means validating static objects like grammars or the process of applying them and, further, whether the latter must be by simulation, not emulation (which is alright for natural language processing). If simulation, moreover, how congitively fine-grained must this be?

Marvin Minsky, at the 2006 AAAI Fellows Symposium to mark the 50th Anniversary of the 1956 Dartmouth Summer Conference on AI, said that Chomsky had blocked the development of computational linguistics for forty years, i.e. from the mid 1960s. I do not agree: computational linguistics, and natural language processing, learnt how to fly by the early 1970s. The question is how and where they are flying now and, here, how they see mainstream linguistics.

An historical excursion

Looking briefly at the history makes the fact of flying clear, and supports my major point.

When computational work on language began in the 1950s, it was primarily on machine translation. The first and most important point about this early work was that the researchers had to struggle specifically to develop process algorithms, most obviously for (syntactic) parsing, as well as to formulate appropriate data structures for dictionaries and grammars. They had also to beat limited machines and inappropriate programming languages into submission. But there were some theoretical ideas about, notably Harris's, that were influential because

they meshed with the predominantly localist approaches to structure definition and translation that researchers then took and also, in the form of distributional views, that supplied a motivation for corpus-based data analysis. Many of those involved in this research were 'traditional' linguists, for example Russian specialists. But there were others who came into computation who were familiar with ideas from logic or philosophy (for instance from Carnap or Wittgenstein).

The net result of all of this was a rich and varied mass of activity well illustrated by conferences like the 1961 Conference on Machine Translation of Languages and Applied Language Analysis, and by the accounts of early research, for example by Plath and Yngve in Booth's 1967 machine translation volume. Moreover, while Chomsky's 'Three models' (1956) and 'Syntactic Structures' (1957) were widely seen as exciting and important, there were other approaches to grammar, for example Dependency Grammar, that offered more than notational variation on Chomsky. There was work on semantics, both for the sense selection that translation cannot ignore and on logical forms to support knowledge representation and reasoning. There were those (like Petrick) sufficiently influenced by Chomsky to go straight for transformational parsing, but this was not a useful computational line and Woods' Augmented Transition Networks were much more productive. As the papers in Rustin's significantly titled *Natural language processing* (1973) showed, computationally motivated and principled work covered a wide range of approaches, functions and tasks, as illustrated by, for example, Kay's MIND system. It would nevertheless be fair to say that for most of the 1960s, Chomskyan linguists and computationl linguists shared a general belief in the importance of a formal apparatus that pairs word strings with grammatical structure descriptions.

But since then there has been a divergence. On the computational side, in spite of Chomskian linguists' concerns and the loss of machine translation funding, research continued and expanded in the 1970s without much input from mainstream linguistics. It had to model process. It built all-embracing task systems, notably in the Speech Understanding Programme, and thus addressed architecture issues. It explored AI-driven approaches focused on the knowledge representation and inference required for task systems, and worked on logical semantics and on pragmatics, both exploiting ideas taken from philosophy (e.g. Lewis, Davidson; Grice). Even the work on grammar types illustrated by LFG, GPSG, and HPSG, most closely tied to linguistics, increasingly developed its own independent momentum. It was not always clear whether, setting aside other games, computational linguistics or natural language processing was the name of the game in all this, but the interaction, or tension, could be fruitful for both.

Thus by the 1980s it was already clear that computational linguistics and natural language processing were advancing without referring significantly to mainstream linguistics or being significantly inadequate thereby. And this has become even more evident in the 1990s with the explosion of corpus-based research and the use of machine learning. This is more than mere fancy data-gathering using statistical surrogates for explanatory models of language behaviour, because it embodies probabilistic ideas which make their own contribution to models of language. Some well-known corpora, like Treebank, have annotations that reflect earlier grammatical concerns, but many corpora now are raw by comparison or are characterised in quite other ways, with other outcomes for what is learnt about language. The growth of corpus-based statistical learning since the 1990s has placed large new objects to consider in the computational linguistic space. These bear questions about statistical approaches to

language: questions about cognitive pertinence; about simulation versus emulation; about granularity in modelling; about the relation between statistical and symbolic if both have a role (as explored in a 1999 Royal Society meeting - see Sparck Jones et al. 2000), whether for individual language 'components' or among components in multi-module systems.

However it is also the case that mainstream, i.e. Chomskyan, linguistics has followed its own path away from its mechanistic 1960s type of language model, towards more generic theories based on concepts that offer insight into fundamentals rather than simply applicable tools: principles and parameters, for example, can be viewed in this light. In the face of this development we cannot expect mainstream linguistics to supply computational linguistics (and certainly not natural language processing) with resources that are readily deployed, though one might make the case for its potentially supplying an encouragement to deeper thought.

Linguistics: so what?

As this historical summary implies, computational linguistics does not *need* mainstream, non-computational linguistics, whether to supply intellectual credibility or to ensure progress. Computational linguistics is not just linguistics with some practically useful but theoretically irrelevant and obfuscating nerdie add-ons. This is not to say that computational linguists can't, and shouldn't, take advantage of linguistics, or at least avoid culpable ignorance where linguists have something to offer. But the boot is now on the other foot, as Martin Kay was already claiming in 1973. He began his paper in the Rustin volume by saying that "For the most part, linguists are unaware of the importance that computers must one day have for their subject." They still are, but that's no skin off our noses: we have more interesting things to do. Thus while we may believe that computational linguistics continues to develop views of language that are worthwhile (whether because valid or just stimulating), as in thinking about process, and that linguistics ought to take notice, computational linguistics is solidly growing in its own right. It is also doing this the better because of the way it has its friendly neighbours, natural language processing, human language technology and the others, that provide a rather demanding apparatus to test and evaluate its ideas.

This is a comforting conclusion. But it's perhaps more than a little arrogant, and we have to be aware of the danger into which the increasing emphasis on corpus-based statistical approaches to language characterisation and processing may lead us. The growth of computational linguistics or, more specifically, natural language *information* processing is increasingly being done by people with a computational rather than linguistic background; machine learning work needs a mathematical, not a linguistic, training. For these people, statistical strategies like Language Modelling have a seductive 'look no hands' aura, with the added attraction of an appeal to technical expertise.

So we need to be alert. It's not just that we may find ourselves putting the cart before the horse. We can get obsessed with the wheels, and finish up with uncritically reinvented, or square, or over-refined or otherwise unsatisfactory wheels, or even just unicycles. What matters is the way the cart, its load, and the horse, together make a rational journey. As the contributors to the Royal Society meeting emphasised, it's about how you integrate the language and the computation on the one hand, the symbolic and the statistical on the other:

and to succeed in this we should not forget that mainstream linguistics may have some things to offer us, even if not as many as linguists themselves may suppose.

My three immediate predecessors in this space reinforce my argument. We have to do computational linguistics and its congeners - natural language processing, etc., - properly. That means, as Annie Zaenen (2006) pointed out, being a lot more analytical about what we are doing in annotation than we often are, and being a lot more cautious than we (or linguists) are about leaning on that wonderful linguistic prop, our native speaker intuitions. It also means, as Adam Kilgariff (2007) argues, being careful about corpora, though retrieval experience shows that a priori purism about what a corpus should be like can be as dangerous as just shovelling up anything. Finally, following Ehud Reiter (2007), just as I maintained computational linguistics ought to be self-confident but not ignore linguistics altogether, of course it must not cut itself off from all the other pertinent areas, like artificial intelligence, that surround it.

Acknowledgement
I am very grateful to Steve Pulman and Yorick Wilks for comments.

References

Booth, A.D. (Ed.) *Machine translation*, Amsterdam: North-Holland, 1967.

Chomsky, N."Three models for the description of language", *IRE Transactions on Information Theory*, Vol. IT-2, 1956, 113-124.

Chomsky, N. *Syntactic structures*, The Hague: Mouton, 1957.

Proceedings of the Conference on Machine Translation of Languages and Applied Language Analysis (1961), London: HMSO, 1962.

Kilgariff, A. "Googleology is bad science", *Computational Linguistics*, 2007.

Reiter, E. "The shrinking horizons of computational linguistics", *Computational Linguistics*, 2007.

Rustin, R. (Ed.), *Natural language processing*, New York: Algorithmics Press, 1973.

Sparck Jones, K., Gazdar, G. and Needham, R.M. (Eds.) "Computers, language and speech: formal theories and statistical data", *Philosophical Transactions of the Royal Society*, Series A, Vol. 358, No. 1769, 2000, 1258-1431.

Zaenen, E. "Markup barking up the wrong tree", *Computational Linguistics*, 32 (4), 2007, 577-580.