Downloaded from rsta.royalsocietypublishing.org

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A MATHEMATICAL PHYSICAL & ENGINEERING SCIENCES

The role of taxonomy in language engineering

Geoffrey Sampson

Phil. Trans. R. Soc. Lond. A 2000 **358**, 1339-1355
doi: 10.1098/rsta.2000.0590

Email alerting service

Receive free email alerts when new articles cite this article - sign up in the box at the top right-hand corner of the article or click **here**

To subscribe to *Phil. Trans. R. Soc. Lond. A* go to:
http://rsta.royalsocietypublishing.org/subscriptions

This journal is © 2000 The Royal Society

Downloaded from rsta.royalsocietypublishing.org

THE ROYAL SOCIETY

MATHEMATICAL, PHYSICAL & ENGINEERING SCIENCES
PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A

The role of taxonomy in language engineering

By Geoffrey Sampson

School of Cognitive and Computing Sciences, University of Sussex, Falmer, Brighton BN1 9QH, UK

To achieve systems that successfully process real-life written and spoken language, a prime need is for taxonomic work that introduces the kind of order and well-definedness into the diversity of linguistic forms and structures that the Linnaean system introduced into the biological realm. The current emphasis on statistically based techniques brings this need into particularly sharp focus. To date, the emphasis within the discipline has tended to be different from this. Computational linguistics has focused more on developing parsing software, for instance, than on defining the target analyses that a parser hits or misses. Enterprises that have discussed standards for language engineering have tended to list sets of approved categories, rather than to specify precise boundaries between categories. There is an analogy with the history of software engineering. Creating software is a relatively glamorous activity, but good software is achieved by postponing the glamour in favour of the plodding work of refining specifications.

Keywords: parsing; natural language processing; language engineering; grammatical taxonomy

MATHEMATICAL, PHYSICAL & ENGINEERING SCIENCES
PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A

1. Software engineering versus programming

The aim of this paper is to suggest that current natural language computing needs to take on board, more fully than it has done up to now, lessons that the wider IT profession learned some 20–30 years ago. The lessons I have in mind were those that led to the creation of the discipline of software engineering. Let me quote historical remarks from two standard textbooks.

> The term 'software engineering' was first introduced in the late 1960s at a conference held to discuss what was then called the 'software crisis'.... Early experience in building large software systems showed that existing methods of software development were not good enough. Techniques applicable to small systems could not be scaled up. Major projects were sometimes years late, cost much more than originally predicted, were unreliable, difficult to maintain and performed poorly. Software development was in crisis. (Quotation taken from Sommerville (1992, p. 3).)

> In the middle to late 1960s, truly large software systems were attempted commercially.... The large projects were the source of the realization that building large software systems was materially different from building small systems.... It was discovered that the problems in building

© 2000 The Royal Society

Downloaded from rsta.royalsocietypublishing.org

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A MATHEMATICAL, PHYSICAL & ENGINEERING SCIENCES

> large software systems were not a matter of putting computer instructions together. Rather, the problems being solved were not well understood, at least not by everyone involved in the project or by any single individual. People on the project had to spend a lot of time communicating with each other rather than writing code. People sometimes even left the project, and this affected not only the work they had been doing but the work of the others who were depending on them. Replacing an individual required an extensive amount of training about the 'folklore' of the project requirements and the system design.... These kinds of problems just did not exist in the early 'programming' days and seemed to call for a new approach. (Quotation taken from Ghezzi *et al.* (1991, p. 4).)

The new approach was what came to be called 'software engineering', which is a fundamental component of the training of computing professionals nowadays.

There are different ways of glossing the term 'software engineering', and I hope my definition will not seem objectionable to readers who are involved with it more centrally than I am, but one way of explaining the concept in a nutshell might be to call it a systematic training of computing professionals in resisting their natural instincts.

For most individuals who are attracted to working with computers, the enjoyable aspect of the work is programming, and running one's programs. Writing code, and seeing the code one has written make things happen, is fun. (It is fun for some people, at any rate; it leaves others cold, but those others will look elsewhere for a career.) Even inserting comments in one's code feels by comparison like a diversion from the real business; programmers do it because they know they should, not out of natural inclination. As for documenting a finished software system on paper, that is real punishment, to be done grudgingly and seeming to require only a fraction of the care and mental effort needed in coding, where every dot and comma count. What is more, these instincts were reinforced in the early years by the instincts of IT managers, who wanted objective ways of monitoring the productivity of the people under them, and quite inevitably saw lines of code per week as a natural measure.

These instincts seem to be widely shared, and they were often harmless in the early years, when software development was on a small scale, more like a craft than an industrial process, and where all the considerations relevant to a particular system might reside in a single head. They led to crisis, once the scale of software projects enlarged, and required teamwork and integrity of software operation under different conditions over long periods of time.

Software engineering addresses that crisis by inverting computing professionals' instinctive scale of values and sequence of activities. Documentation, the dull part, becomes the central and primary activity. Developing a software system becomes a process of successively developing and refining statements on paper of the task and intended solution at increasing levels of detail—requirements definitions, requirements specifications, software specifications—so that the programming itself becomes the routine bit done at the end, when code is written to implement specifications of such precision that, ideally, the translation should be more or less mechanical; conceptual unclarities that could lead to faulty program logic should be detected and

Downloaded from rsta.royalsocietypublishing.org

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A MATHEMATICAL, PHYSICAL & ENGINEERING SCIENCES

eliminated long before a line of code is written. Weinberg (1971) argued for a culture of 'egoless programming', which systematically deprives computing professionals of the pleasures of individual creativity and control over the programs for which they are responsible, as a necessary price to be paid for getting large systems that work as wholes.

Nobody suggests that now we have software engineering, all the problems described 30 years ago as 'software crisis' have melted away and everything in the software development garden is rosy. But I think few people in the IT industry would disagree that the counter-instinctive disciplines of software engineering are a necessary condition for successful software development, though those disciplines are often difficult to apply, and clearly they are not sufficient to ensure success.

2. How far we have come

Natural language computing is not a new application of computer technology. When Alan Turing drew up a list of potential uses for the stored-program electronic computer, a few weeks after the world's first computer run at Manchester in June 1948, the second and third items on his five-item list were 'learning of languages' and 'translation of languages' (Hodges 1983, p. 382). Some of the early machine translation projects must have been among the larger software-development projects in any domain in the 1950s and early 1960s. On the whole, however, natural language computing has been late in making the transition from individualistic, craft activity to industrial process, and, where work was being done in a more realistic style, for instance on Peter Toma's 'Systran' machine-translation system (Hutchins & Somers 1992, ch. 10), for many years it was given the cold shoulder by computational linguists within the academic world (Sampson 1991, pp. 127–128).

Since the 1980s, in some respects the subject has made great strides in the relevant direction. It is hard, nowadays, to remember the cloistered, unrealistic ethos of natural language computing as it was less than 20 years ago. To give an impression of how things were then, let me quote below (as I have done elsewhere) a typical handful of the language examples used by various speakers at the inaugural meeting of the European Chapter of the Association for Computational Linguistics, held at Pisa in 1983, in order to illustrate the workings of the various software systems that the speakers were describing.

(i) *Whatever is linguistic is interesting.*

(ii) *A ticket was bought by every man.*

(iii) *The man with the telescope and the umbrella kicked the ball.*

(iv) Hans bekommt von dieser Frau ein Buch.

(v) *John and Bill went to Pisa. They delivered a paper.*

(vi) Maria é andata a Roma con Anna.

(vii) *Are you going to travel this summer? Yes, to Sicily.*

Downloaded from rsta.royalsocietypublishing.org

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A MATHEMATICAL, PHYSICAL & ENGINEERING SCIENCES

Some critics of the field were unwilling to recognize such material as representing human language at all. Michael Lesk (now Director, Information and Intelligent Systems, at the US National Science Foundation) once characterized it acidly as 'the imaginary language, sharing a few word forms with English, that is studied at MIT and some other research institutes' (Lesk 1988). To me, there was nothing wrong with these dapper little example sentences as far as they went; but they were manifestly invented rather than drawn from real life, and they were invented in such a way as to exclude all but a small fraction of the problematic issues that confront software that attempt to deal with real-life usage. Focusing on such artificial examples gave a severely distorted picture of the issues facing natural language engineering. Contrast the above examples with, at the other extreme, a few typical utterances taken from the structurally annotated CHRISTINE corpus,† which my research group released this summer, based on real-life material extracted from the British National corpus:‡

(i) *well you want to nip over there and see what they come on on the roll*

(ii) *can we put erm New Kids # no not New Kids Wall Of # you know*

(iii) *well it was Gillian and # and # erm* {pause} *and Ronald's sister erm* {pause} *and then er* {pause} *a week ago last night erm* {pause} *Jean and I went to the Lyceum together to see* Arsenic and Old Lace

(iv) *lathered up, started to shave* {unclear} {pause} *when I come to clean it there weren't a bloody blade in, the bastards had pinched it*

(v) *but er* {pause} *I don't know how we got onto it* {pause} *er sh- # and I think she said something about oh she knew her tables and erm* {pause} *you know she'd come from Hampshire apparently and she #* {pause} *an- # an- yo- # you know er we got talking about ma- and she's taken her child away from* {pause} *the local school* {pause} *and sen- # is now going to a little private school up* {pause} *the Teign valley near Teigngrace apparently fra-*

Whatever IT application we have in mind, whether automatic information extraction, machine translation, generation of orthographically conventional typescript from spoken input, or something else, I think that the degree of complexity and difficulty presented by the second set of examples, compared with the first set, is quite manifest.

Of course, I have made the point vivid by using examples drawn from spontaneous, informal speech (but then, notice that the last, at least, of the examples quoted from the Pisa meeting was clearly intended to represent speech rather than writing). Some natural language computing applications are always going to relate to written language rather than speech, and writing does tend to be more neatly regimented than the spoken word. Even published writing, however, after authors and editors have finished redrafting and tidying it, contains a higher incidence of structural unpredictability and perhaps anarchy than the examples from the Pisa conference. A few sentences drawn at random from the LOB corpus¶ of published British English follow.

† For more information, visit http://www.cogs.susx.ac.uk/users/geoffs/RChristine.html.
‡ For more information, visit http://info.ox.ac.uk/bnc/.
¶ For more information, visit http://www.hit.uib.no/icame.html.

Downloaded from rsta.royalsocietypublishing.org

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A MATHEMATICAL, PHYSICAL & ENGINEERING SCIENCES

(i) *Sing slightly flat.*

(ii) *Mr. Baring, who whispered and wore* pince-nez, *was seventy if he was a day.*

(iii) *Advice – Concentrate on the present.*

(iv) *Say the power-drill makers, 75 per cent of major breakdowns can be traced to neglect of the carbon-brush gear.*

(v) *But he remained a stranger in a strange land.*

In the first example we find a word in the form of an adjective, *flat*, functioning as an adverb. In the next example, the phrase *Mr. Baring* contains a word ending in a full stop followed by a word beginning with a capital, which, exceptionally, does not mark a sentence boundary. The third 'sentence' links an isolated noun with an imperative construction in a logic that is difficult to pin down. In *Say the power-drill makers*..., verb precedes subject for no very clear reason. The last example is as straightforward as the examples from the Pisa meeting; but, even in traditional published English, straightforward examples are not the norm. (Currently, technologies such as email are tending to make written language more like speech.)

There were no technical obstacles to real-life material of this sort being used much earlier than it was. The Brown corpus† of American English, which is proving to be a very valuable research resource even now after the close of the 20th century, was published as early as 1964; for decades it was all but ignored.

For computational linguists to develop software systems based entirely on well-behaved invented data, which was the norm throughout the 1980s, is analogous to the home-computer buff who writes a program to execute some intellectually interesting function, but has little enthusiasm for organizing a testing regime that would check the viability of the program by exposing it to a realistically varied range of input conditions. And this approach to natural language computing militates against any application of statistical processing techniques. Speakers of a natural language may be able to make up example sentences of the language out of their heads, but they certainly cannot get detailed statistical data from their intuitions.

One must learn to walk before one runs, and the 1980s reliance on artificial linguistic data might be excused on the grounds that it is sensible to begin with simple examples before moving on to harder material. In fact, I think that the preference of the discipline for artificial data went much deeper than that. In the first place, as we have seen, computational linguistics was not 'beginning' in the 1980s. More important, almost everyone involved with linguistics was to a greater or lesser extent under the spell of the immensely influential American intellectual Noam Chomsky of the MIT, who saw linguistics as more an aprioristic than an empirical discipline.

One of Chomsky's fundamental doctrines was his distinction between linguistic 'performance'—people's observable, imperfect linguistic behaviour—and linguistic 'competence', the ideal, intuitively accessible mental mechanisms that were supposed to underlie that performance (Chomsky 1965, p. 4). Chomsky taught that the subject worthy of serious academic study was linguistic competence, not performance. The route to an understanding of linguistic performance could lie only through prior

† For more information, visit http://www.hit.uib.no/icame.html.

Downloaded from rsta.royalsocietypublishing.org

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A MATHEMATICAL, PHYSICAL & ENGINEERING SCIENCES

analysis of competence (Chomsky 1965, pp. 9, 15), and the tone of Chomsky's discussion did not encourage his readers to want to move on from the latter to the former.

For Chomsky, this concept of an ideal linguistic competence residing in each speaker's mind was linked to his (thoroughly misguided) idea that the detailed grammatical structure of natural languages is part of the genetic inheritance of our species, like the detailed structure of our anatomy.† However, Chomsky was successful in setting much of the agenda of linguistics even for researchers who had no particular interest in these psychological or philosophical questions. In consequence, if computational linguists of the 1980s noticed the disparity between the neatly regimented examples used to develop natural language processing software and the messy anarchy of real-life usage, rather than seeing that as a criticism of the examples and the software, they tended obscurely to see it as a criticism of real-life usage. Aarts & Van den Heuvel (1985) give a telling portrayal of the attitudes that were current in those years. Not merely did most natural language computing not use real-life data, but for a while there seemed to be an air of mild hostility or scorn towards the minority of researchers who did.

Happily, from about 1990 onwards the picture has completely changed. Over the last ten years it has become routine for natural language computing research to draw on 'corpora', machine-readable samples of real-life linguistic usage; and the validity of statistics-based approaches to natural language analysis and processing is now generally accepted. I am not sure that one can count this as a case of the profession being convinced by the weight of reasoned argument; my impression of what happened was that American research-funding agencies decided that they had enough of natural language computing in the aprioristic style and used the power of the purse to impose a change of culture, which then spread across the Atlantic, as things do. But, however it came about, the profession has now accepted the crucial need to be responsive to empirical data.

3. The lesson not yet learned

In another respect, though, it seems to me that natural language computing has yet to take on board the software-engineering lesson of the primacy of problem analysis and documentation over coding.

I shall illustrate the point from the field of parsing: automatic grammatical analysis of natural language. I believe similar things could be said about other areas of natural language processing; but I am a grammarian myself, and automatic parsing is a key technology in natural language computing. Many would have agreed with K. K. Obermeier's assessment ten years ago that parsing was 'The central problem' in virtually all natural language processing applications (Obermeier 1989, p. 69); more recently, I notice that 'parsing' takes up more space than any other technology name in the index of an NSF/European Commission-sponsored survey of natural language and speech computing (Cole *et al.* 1997).‡ As these pointers suggest, a large

† Chomsky expounded this doctrine of linguistic nativism in books such as *Reflections on language* (Chomsky 1976); it has been popularized recently by Steven Pinker's *The language instinct* (Pinker 1994). I have pointed out the vacuousness of Chomsky's and Pinker's various arguments for linguistic nativism in my *Educating Eve* (Sampson 1997).

‡ Only general concepts such as *corpora*, *dialogue*, *speech*, *word* occupy larger sections of Cole *et al.*'s (1997) index.

Downloaded from rsta.royalsocietypublishing.org

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A MATHEMATICAL, PHYSICAL & ENGINEERING SCIENCES

number of research groups worldwide have been putting a lot of effort into solving the parsing problem for years and, indeed, for decades. Many parsing systems have been developed, using different analytic techniques and achieving different degrees of success.

Any automatic parser is a system that receives as input a representation of a spoken or written text, as a linear sequence of words (together possibly with subsidiary items, such as punctuation marks in the case of written language), and outputs a structural analysis, which is almost always in a form notationally equivalent to a tree structure, having the words of the input string attached to its successive leaf nodes, and with non-terminal nodes labelled with grammatical categories drawn from some agreed vocabulary of grammatical classification. (A minority of research groups working on the parsing problem use output formalisms that deviate to a certain extent from this description, for instance the 'dependency' notation due to Tesnière (1959), but I do not think these differences are significant enough to affect the substance of the point I am developing.) The structural analysis is something like a representation of the logic of a text, which is physically realized as a linear string of words because the nature of speech forces a one-dimensional linear structure onto spoken communication (and writing mimics the structure of spoken utterances). So it is easy to see why any automatic processing that relates to the content of spoken or written language, rather than exclusively to its outward form, is likely to need to recover the tree-shaped structures of grammar underlying the string-shaped physical signals.

Obviously, to judge the success of any particular parser system, one must not only see what outputs it yields for a range of inputs, but must know what outputs it *should* produce for those inputs: one must have some explicit understanding of the target analyses, against which the actual analyses can be assessed. Yet it was a noticeable feature of the literature on automatic natural language parsing for many years that, while the software systems were described in detail, there was hardly any public discussion of the schemes of analysis that different research groups were treating as the targets for their parsing systems to aim at. Issues about what counted as the right analyses for particular input examples were part of what Ghezzi *et al.* (1991) called 'the "folklore" of the project requirements'. Members of particular parsing projects must have discussed such matters among themselves, but one almost never saw them spelled out in print.

Of course, unlike some of the topics that software is written to deal with, natural language parsing is a subject with a long tradition behind it. A number of aspects of modern grammatical analysis go back 2000 years to the Greeks; and the idea of mapping out the logic of English sentences as tree structures was a staple of British schooling at least 100 years ago. So computational linguists may have felt that it was unnecessary to be very explicit about the targets for automatic parsing systems, because our shared cultural inheritance settled that long since.

If people did think that, they were wrong. The wrongness of this idea was established experimentally, at a workshop held in conjunction with the Association of Computational Linguistics annual conference at Berkeley, California, in 1991. Natural language processing researchers from nine institutions were each given the same set of English sentences and asked to indicate what their respective research groups would regard as the target analyses of the sentences, and the nine sets of analyses were compared. These were not particularly complicated or messy sentences; they

Downloaded from rsta.royalsocietypublishing.org

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A MATHEMATICAL, PHYSICAL & ENGINEERING SCIENCES

were drawn from real-life corpus data, but as real-life sentences go they were rather well-behaved examples. And the comparisons were not made in terms of the labels of the constituents: the only question that was asked was how far the researchers agreed on the shapes of the trees assigned to the sentences; that is, to what extent they identified the same subsequences of words as grammatical constituents, irrespective of how they categorized the constituents they identified.

The level of agreement was strikingly low. For instance, only the two subsequences marked by square brackets were identified as constituents by all nine participants in the following example (and results for other cases were similar).

One of those capital-gains ventures, in fact,
has saddled him [with [Gore Court]].

If specialists agree as little as this on the details of what parsing systems are aiming to do, that surely establishes the need for a significant fraction of all the effort and resources that are put into automatic parsing to be devoted to discussing and making the targets which the software is aiming at more publicly explicit, rather than putting them all into improving the software.

4. The scale of the task

I do not mean to imply that every natural language computing group working on English ought to agree on a single common parsing scheme. In the context of applications executing commercially or socially valuable natural language processing functions of various kinds, automatic parsing is only a means to an end. It may well be that the kind of structural analysis that is most appropriate with respect to one function differs in some details from the analysis that is appropriate for an application executing a different function. But the lack of agreement revealed at the 1991 workshop did not arise, because various research groups had made explicit decisions to modify the details of a recognized public scheme of English-language parsing to suit their particular purposes. No such public scheme existed. Separate groups were forced to use different parsing schemes, because each research group had to develop its own standards, as a matter of internal project 'folklore'. The analytic concepts that we inherit from traditional school grammar teaching may be fine as far as they go, but they are far too limited to yield unambiguous, predictable structural annotations for the myriad linguistic constructions that occur in real life.

Additionally, because research groups developed their parsing standards independently and in an informal fashion, not perceiving this as truly part of the work they were engaged in, they were in no position to develop schemes that were adequate for the massive structural complexity of any natural language. The results of the 1991 ACL workshop experiment came as little surprise to me, in view of earlier experiences of my own. From 1983 onwards, as a member of the University of Lancaster natural language computing group, I took responsibility for creating a written-English 'treebank': a sample of structurally annotated real-life sentences,† which was needed as a resource for a statistics-based parsing project led by my senior colleague Geoffrey Leech. I remember that when I took the task on and we needed to agree an

† I believe that the term 'treebank' was first coined in this sense by Geoffrey Leech in connection with our Lancaster project. It has subsequently become current internationally.

Downloaded from rsta.royalsocietypublishing.org

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A MATHEMATICAL, PHYSICAL & ENGINEERING SCIENCES

annotation scheme for the purpose, Leech (who knows more about English grammar than I ever shall) produced a 25-page typescript listing a set of symbols he proposed that we use, with guidelines for applying them in debatable cases; and I thought this represented such a thorough job of anticipating problematic issues that it left little more to be said. All I needed to do was to use my understanding of English in order to apply the scheme to a series of examples.

I soon learned. As I applied the scheme to a sample of corpus data, the second or third sentence I looked at turned out to involve some turn of phrase that the typescript did not provide for; as I proceeded, something on the order of every other sentence required a new annotation precedent to be set. Written items like names, addresses, money sums, weights and measures have linguistic structure of their own; the grammatical tradition says little about them, so one has to make new decisions about how to represent that structure. However, plenty of decisions are needed also in the more linguistically 'central' areas of clause and phrase analysis. Often, alternative structural annotations of a given construction each seemed perfectly defensible in terms of the school grammatical tradition, but if we were going to use our treebank to produce meaningful statistics, we had to pick one alternative and stick to it.

Consider, to give just one example, the construction exemplified in *the more, the merrier*; the construction that translates into German with *je* and *desto*. Here are three ways of grouping a sentence using that construction into constituents:

(i) [[*the wider the wheelbase is*], [*the more satisfactory is the performance*]]

(ii) [[*the wider the wheelbase is*], *the more satisfactory is the performance*]

(iii) [[[*the wider the wheelbase is*], *the more satisfactory*] *is the performance*]

The two clauses might be seen as coordinated, as in the first line, since both have the form of main clauses and neither of them contains an explicit subordinating element. Or the second clause might be seen as the main clause, with the first as an adverbial clause adjunct. Or the first clause might be seen as a modifier of the adjectival predicate within the second clause. There seemed to be no strong reason to choose one of these analyses rather than another.

Linguists influenced by the concept of innate psychological 'competence' tend to react to alternatives like this by asking which analysis is 'true' or 'psychologically real'; which structure corresponds to the way the utterance is processed by speaker's or hearer's mental machinery. However, even if questions like that could ultimately be answered, they are not very relevant to the tasks confronting natural language computing here and now. We have to impose analytic decisions in order to be able to register our data in a consistent fashion; we cannot wait for the outcome of abstruse future psychological investigations.

Indeed, I should have thought it was necessary to settle on an analytic framework in order to adequately assemble comprehensive data for the theoretical psycholinguists to use in their own investigations. In biology, the Linnaean binomial classification system (which Linnaeus and everyone else knew to be unnatural, but was practical to apply) was a prior requirement for the development of modern theories of cladistics. A science is not likely to be in a position to devise deep theories to explain its data, before it has an agreed scheme for identifying and registering those data. To use the terms 'true' and 'false' in connection with a scheme of grammatical annotation would

Downloaded from rsta.royalsocietypublishing.org

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A MATHEMATICAL, PHYSICAL & ENGINEERING SCIENCES

be as inappropriate as asking whether the alphabetical order from A to Z that we use for arranging names in a telephone directory or books on shelves is the 'true' order.

At any rate, within the Lancaster group it became clear that our approach to automatic parsing, in terms of seeking structures over input word-strings that conformed to the statistics of parse configurations in a sample of analysed material, required us to evolve far more detailed analytic guidelines than anything that then existed; without them, the statistics would be meaningless, because separate instances of the same construction would be classified now one way, now another. We evolved a routine in which each new batch of sentences manually parsed would lead to a set of tentative new analytic precedents that were logged on paper and circulated among the research team; weekly or fortnightly meetings were held where the new precedents were discussed and either accepted or modified, for instance because a team member noticed a hidden inconsistency with an earlier decision. The work was rather analogous to the development of the Common Law. A set of principles attempts to cover all the issues on which the legal system needs to provide a decision, but human behaviour continually throws up unanticipated cases for which the existing legal framework fails to yield an unambiguous answer; so new precedents are set, which cumulatively make the framework increasingly precise and comprehensive. We want our nation's legal system to be consistent and fair, but perhaps above all we want it to be fully explicit; and if that is possibly not the dominant requirement for a legal system, it surely is for a scientific system of data classification. To quote Jane Edwards of the University of California at Berkeley: 'The single most important property of any data base for purposes of computer-assisted research is that *similar instances be encoded in predictably similar ways*' (Edwards 1992, p. 139).

Ten years of our accumulated precedents on structural annotation of English turned a 25-page typescript into a book of 500 large-format pages (Sampson 1995). Beginning from a later start, the Pennsylvania treebank group published their own independent but very comparable system of structural annotation guidelines on the Web in the same year (Bies *et al*. 1995). I am sure that the Pennsylvania group feel as we do, that neither of these annotation schemes can be taken as a final statement; the analogy with the growth of the law through cumulation of precedents suggests that there never could be a last word in this domain. My own group has been elaborating our scheme in the last few years by applying it to spontaneous speech; but although the main focus here is on aspects of the annotation scheme that were irrelevant to the structure of written prose, for instance mechanisms for marking what is going on when speakers edit their utterances 'on the fly', we continue to find ourselves setting new precedents for constructions that are common to writing as well as speech. (In due course, we plan to cumulate them into a supplement to the 1995 book.)

5. Differential reception of data and specifications

The only way that one can produce an adequate scheme of structural annotation is to apply an initial scheme to real data and refine the scheme in response to problem cases, as we have been doing; so, in developing an annotation scheme one inevitably generates a treebank, an annotated language sample, as a byproduct. The Lancaster treebank that started me on this enterprise in the mid-1980s was for internal project use and was never published, but I did electronically publish the annotated samples on which later stages of annotation-scheme development were based. This

Downloaded from rsta.royalsocietypublishing.org

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A MATHEMATICAL, PHYSICAL & ENGINEERING SCIENCES

'SUSANNE corpus', as it is called, was released in successively more accurate versions between 1992 and 1994. Part of the point I am seeking to make in the present paper can be illustrated by the different receptions accorded by the research community to the SUSANNE corpus, and to the published definition of the SUSANNE annotation scheme.

Because it emerged from a manual annotation process that aimed to identify and carefully weigh up every debatable analytic issue arising in its texts, the SUSANNE corpus is necessarily a small treebank; there is a limit to how reliable any statistics derived from it can hope to be. Yet it has succeeded far beyond my expectations in establishing a role for itself internationally as a natural language computing research resource. Accesses to the ftp site originally distributing it at the Oxford Text Archive quickly rose to a high level (and subsequently other 'mirror' sites began distributing it, so that I no longer have any way of monitoring overall accesses). I quite often encounter in the professional literature references to research based on the SUSANNE corpus, commonly by researchers of whom I had no prior knowledge.

Conversely, the book defining the annotation scheme has found no role that I have detected. Reviewers have made comments that were pleasing to read, but almost no-one has spontaneously found reasons to get into correspondence about the contents of the annotation scheme, in the way that many researchers have about the SUSANNE treebank; indeed, it often becomes apparent, when people who have been working intensively with the SUSANNE corpus get in touch, that they have never looked at the published definition of the SUSANNE annotation scheme on which the corpus is based. My guess is that the only place where that reference book is in routine day-to-day use is in my own research group at Sussex.†

Now, like every academic, I am naturally quite delighted to find that any research output for which I was responsible seems to be meeting a need among the international research community. The welcome that the corpus alone has received is certainly more than a sufficient professional reward for the effort that created corpus and annotation scheme. Nevertheless, I find the imbalance in the reception of the two resources rather regrettable in what it seems to say about the values of the discipline. In my own mind, the treebank is an appendix to the annotation scheme, rather than the other way round; the treebank serves a function similar to what I believe biologists call a type collection attached to a biological taxonomy: a set of specimens intended to clarify the definitions of the taxonomic classes. The SUSANNE treebank is really too small to count as a significant database of English grammatical usage; whereas the published annotation scheme, although it unquestionably has many serious limitations and imperfections, can (I believe) claim to be a more serious attempt to do its own job than anything that existed in print before. If the research community is not taking up the SUSANNE annotation scheme as a basis from which to push forward the enterprise of taxonomizing English structure, that could just mean that they prefer the Pennsylvania scheme as a starting point for that work; but in fact I do not get the impression that this sort of activity has been getting under way in connection with the Pennsylvania scheme either. (The fact that the Pennsylvania group limited themselves to publishing their scheme via the Web rather than as a printed book perhaps suggests that they did not expect it to.)

† One of the organizers of The Royal Society Discussion Meeting informs me that this guess is not wholly correct. Nevertheless, the difference between reception of the scheme and reception of the data is striking.

Downloaded from rsta.royalsocietypublishing.org

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A MATHEMATICAL, PHYSICAL & ENGINEERING SCIENCES

When Geoffrey Leech began to look for support to create the first corpus of British English, about 30 years ago, I understand that funding agencies were initially unreceptive, because at that time a simple collection of language samples did not strike reviewers as a valid research output. People expected concrete findings, not just a collection of data from which findings could subsequently be generated; although Leech's LOB Corpus, when it was eventually published in 1978, served as the raw material for a huge variety of research findings by many different researchers, which collectively must far exceed the new knowledge generated by almost any research project that seeks to answer a specific scientific question.

We have won that battle now, and it is accepted that the compilation of natural language corpora is a valuable use of research resources; though now that massive quantities of written language are freely available via the Internet, the need at the beginning of the new century is for other sorts of language sample, representing speech rather than writing and/or embodying various categories of annotation. But there is still a prejudice in favour of the concrete. When I put together a new research proposal, I couch it in terms of compiling a new annotated corpus, rather than extending and testing a scheme of structural annotation. If I wrote the proposals in the latter way, I am convinced they would fail, whereas research agencies are happy to sponsor new corpora even though (given our method of working) the ones I can offer to create are very small. Before software engineering brought about a change of vision, IT managers measured their colleagues' output in terms of lines of code, and overlooked the processes of planning, definition, and coordination that were needed before worthwhile code could be written. At present, most computational linguists see the point of an annotated corpus, but few see the point of putting effort into refining schemes of annotation.

Some encouragement to give more priority to the annotation-scheme development task has come from the (perhaps unexpected) direction of the European Commission, whose Directorate-General XIII induced the predominantly US-sponsored Text Encoding Initiative† to include a small amount of work on this area about 10 years ago, and more recently established the EAGLES group‡ to stimulate the development of standards and guidelines for various aspects of natural language computing resources, including structural annotation of corpora.

The EAGLES initiative has produced valuable work, notably in the area of speech systems, where the relevant working group has assembled, between hard covers, what looks to me like a very complete survey of problems and best practices in various aspects of speech research (Gibbon *et al.* 1997). But in the area of grammatical annotation, the EAGLES enterprise was hobbled by the obvious political necessity for EU-funded work to deal jointly with a large number of European languages, each of which has its own structure, and which are very unequal in the extent to which they have been worked over by either traditional or computer-oriented scholarly techniques (many of them lagging far behind English in that respect). Consequently, in this domain the EAGLES initiative focused on identifying categories that are common to all or most EU national languages, and I think it is fair to say that its specific recommendations go into even less detail than the inherited school grammar

† For more information, visit http://www-tei.uic.edu/orgs/tei/.

‡ The 'Expert Advisory Group on Language Engineering Standards', http://www.ilc.pi.cnr.it/EAGLES96/intro.html.

Downloaded from rsta.royalsocietypublishing.org

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A MATHEMATICAL, PHYSICAL & ENGINEERING SCIENCES

tradition provides for English. The nature of the EAGLES enterprise meant that it could hardly have been otherwise.

What is needed is more effort devoted to identifying and systematically logging the fine details of spoken and written language structure, so that all aspects of our data can be described and defined in terms that are meaningful from one site to another, and this has to be done separately for any one language in its own terms (just as the taxonomy of one family of plants is a separate undertaking from the taxonomy of any other family). European languages obviously do share some common structural features because of their common historical origins and subsequent contacts; but a language adapts its inherited stock of materials to new grammatical purposes on a time-scale of decades—think, for instance, of the replacement of *might* by *may* in the most respectable written contexts just within the last 10 or 20 years, in constructions like *if he had been in Cornwall he may have seen the eclipse*—whereas the EU languages have developed as largely independent systems for millennia. We do not want our grammatical classification systems to be excessively dominated by ancient history.

In developing predictable guidelines for annotating the structure of spontaneous spoken utterances, my group faced large problems stemming from the fact that, within English, there are different groups of speakers who, for instance, use the verb system in different ways. If a speaker of a non-standard version of English says *she done it*, rather than *she did it* or *she's done it* (which speakers very often do), to a schoolteacher, this may represent heresy to be eradicated, but for us it is data to be logged. We have to make a decision about whether such cases should be counted as simple past forms with non-standard use of *done* rather than *did* as past tense of *do*; perfective forms with non-standard omission of the auxiliary; or a third verbal category, alongside the perfective and simple past categories of the standard language. The idea of developing guidelines at this level of detail that simultaneously take into account what happens in German or Modern Greek is really a non-starter.

In any case, encouragement from national or supranational government level will not achieve very much, unless enthusiasm is waiting to be kindled at grass-roots level among working researchers. Natural language computing researchers need to see it as just as fascinating and worthwhile a task to contribute to the identification and systematic classification of distinctive turns of phrase as to contribute to the development of language-processing software systems; so that taxonomizing language structure becomes an enterprise for which the discipline as a whole takes responsibility, in the same way as biologists recognize systematics as an important subfield of their discipline. The fact that natural language computing is increasingly drawing on statistical techniques, which, by their nature, require large quantities of material to be registered and counted in a thoroughly consistent fashion, makes the task of defining and classifying our data even more crucial that it was before. It is surely too important to leave in the hands of isolated groups in Sussex or Pennsylvania.

6. The fractal analogy

If people are attracted to the task, there is plenty of work for them to do. Richard Sharman, of SRI International, Cambridge, has likened natural languages to fractal objects, in the sense that there is always more structural detail to be revealed as one looks at them more closely. I carried out a statistical analysis on the first treebank

Downloaded from rsta.royalsocietypublishing.org

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A MATHEMATICAL, PHYSICAL & ENGINEERING SCIENCES

that we developed at Lancaster in the mid-1980s that suggested that this analogy may be rather exact.† For one high-frequency category of phrase (the noun phrase), I looked at the frequencies of each alternative realization in the treebank. That is, I listed the various sequences of daughter labels found on nodes immediately dominated by a mother node labelled 'noun phrase', using a fairly coarse alphabet of grammatical category labels, and I counted the number of times each distinct daughter-label sequence recurred in the data. The total size of the treebank was not large, for the reasons I have already discussed, but, within the limits of that dataset, there proved to be a rather precise relationship between numbers and frequencies of different constructions (that is, different ways of realizing the given mother category as a sequence of daughter categories). As one looked at constructions with lower and lower frequencies, the number of different constructions each occurring at those frequencies grew, in a regular way, so that constructions that were individually very rare were collectively quite common. Putting it formally:

> if m is the frequency of the commonest single construction (in my data, m was about 28 per 1000 words),
>
> and f is the relative frequency of some construction (fm is its absolute frequency),
>
> then the proportion of all construction tokens that represent construction types of relative frequency less than or equal to f is about $f^{0.4}$.

The significance of this finding is that it contradicts the widely shared picture of a natural language as containing a limited number of 'competent' grammatical structures, which in practice are surrounded by a penumbra of more-or-less random, one-off 'performance errors'; and it is this picture, I believe, that has done much to discourage linguists (even many linguists who would claim to disagree with Noam Chomsky's theories) from seeing structural taxonomy as a worthwhile activity.

If the competence versus performance picture were correct, then identifying the 'competent' constructions would be a task more akin to defining the syntax of Pascal or Java than to classifying the genera and species of the family *Compositae*; and investigating the 'performance errors' would be a fairly unattractive task to most researchers: mistakes are mistakes. But that picture seems to imply that construction frequencies ought to be distributed bimodally, with competent constructions occurring at reasonably high frequencies, individual performance errors each occurring at a low frequency, and not much in-between. My data were not like that; constructions were distributed smoothly along the scale of frequencies, with no striking gaps. (It would in any case be surprising to find that 'performance error' was an important factor in this treebank, which was based on published writing.)

Consider what the mathematical relationship I have quoted would mean, if it continued to hold true in much larger datasets than I was in a position to investigate. (I cannot know that it does, but as far as they went, my data contained no suggestion that the relationship broke down at the lower frequencies.) If the relationship held true in larger bodies of data, then:

† The analytic findings are presented in Sampson (1987); see Sampson (1992, pp. 440–445) for discussion of a critique of my conclusions by Briscoe and others.

Downloaded from rsta.royalsocietypublishing.org

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A MATHEMATICAL, PHYSICAL & ENGINEERING SCIENCES

(i) one in every 10 construction tokens would represent a type that occurs at most once per 10 000 words;

(ii) one in every 100 construction tokens would represent a type that occurs at most once per 3.5 million words;

(iii) one in every 1000 construction tokens would represent a type that occurs at most once per billion words.

One per cent of all construction tokens is surely far too high a proportion of language material for natural language computing to wash its hands of as too unusual to deal with. One might feel that one in a thousand is still too high a proportion for that treatment. Yet, if we have to search millions, or even hundreds of millions, of words of text to find individual examples, we seem to be a long way away from the picture of natural language grammars as limited sets of rules, like programming languages with a number of irregularities and eccentric features added. A natural fractal object such as a coastline can never be fully described, one has to be willing to ignore detail below some cut-off. But the degree of detail that natural language computing ought to be taking into account extends well beyond the range of structures described in standard grammars of English.

7. Conclusion

At the end of the 20th century, humankind's honeymoon with the computer has not yet quite faded, and software development still has a glamour that is lacking in research that revolves round ink and paper. But a computational linguist who helps to develop a natural language software system is devoting himself to a task that, realistically, is unlikely to achieve more than a tiny advance on the current state of the art, and will quickly be forgotten when another, better system is produced. To improve our system for registering and classifying the constructions of English, on the other hand, is to make a potentially lasting contribution to our knowledge of the leading medium of information storage and exchange on the planet. Of course, I do not suggest that computational linguists should migrate *en masse* from the former activity to the latter, but it would be good to see more of a balance.

Some researchers perhaps feel that the kind of detailed study of natural language structure that I have been advocating belongs to the domain of the humanities rather than science, and, consequently, is not for them. But arts-based researchers are not inclined towards the work of imposing rigid and sometimes artificial classificatory divisions on inherently continuous clines, which is needed as a prerequisite for gathering large amounts of quantitative data on a consistent basis. The flavour of the work makes it more akin to other IT activities than to the humanities, and it is computer-oriented researchers who have a motive for engaging in it. I hope that more of them will begin to do so.

References

Aarts, J. & van den Heuvel, T. 1985 Computational tools for the syntactic analysis of corpora. *Linguistics* **23**, 303–335.

Bies, A., Ferguson, M., Katz, K. & MacIntyre, R. 1995 Bracketing guidelines for treebank II style. Available at http://www.cis. upenn.edu/~treebank/home.html.

Downloaded from rsta.royalsocietypublishing.org

Chomsky, A. N. 1965 *Aspects of the theory of syntax*. Cambridge, MA: MIT Press.

Chomsky, A. N. 1976 *Reflections on language*. London: Temple Smith.

Cole, R., Mariani, J., Uszkoreit, H., Varile, G. B., Zaenen, A., Zampolli, A. & Zue, V. (eds) 1997 *Survey of the state of the art in human language technology*. Cambridge University Press.

Edwards, J. 1992 Design principles in the transcription of spoken discourse. In *Directions in corpus linguistics: proceedings of Nobel symposium 82* (ed. J. Svartvik), pp. 129–144. Berlin: Mouton de Gruyter.

Ghezzi, C., Jazayeri, M. & Mandrioli, D. 1991 *Fundamentals of software engineering*. Englewood Cliffs, NJ: Prentice-Hall.

Gibbon, D., Moore, R. & Winski, R. (eds) 1997 *Handbook of standards and resources for spoken language systems*. Berlin: Mouton de Gruyter.

Hodges, A. 1983 *Alan Turing: the enigma of intelligence*. Hutchinson. (My page reference is to the Unwin Paperbacks edition, 1985.)

Hutchins, W. J. & Somers, H. L. 1992 *An introduction to machine translation*. Academic.

Lesk, M. 1988 Review of 'The computational analysis of English' (ed. R. G. Garside, G. N. Leech & G. R. Sampson). *Comp. Ling.* **14**, 90–91.

Obermeier, K. K. 1989 *Natural language processing technologies in artificial intelligence: the science and industry perspective*. Chichester, UK: Ellis Horwood.

Pinker, S. 1994 *The language instinct: the new science of language and mind* (Penguin edition, 1995). New York: William Morrow.

Sampson, G. R. 1987 Evidence against the 'grammatical'/'ungrammatical' distinction. In *Corpus linguistics and beyond* (ed. W. Meijs), pp. 219–226. Amsterdam: Rodopi.

Sampson, G. R. 1991 Natural language processing. In *Humanities research using computers* (ed. C. Turk), ch. 8. London: Chapman & Hall.

Sampson, G. R. 1992 Probabilistic parsing. In *Directions in corpus linguistics: proceedings of Nobel symposium 82* (ed. J. Svartvik), pp. 425–447. Berlin: Mouton de Gruyter.

Sampson, G. R. 1995 *English for the computer: the SUSANNE corpus and analytic scheme*. Oxford: Clarendon.

Sampson, G. R. 1997 *Educating Eve: the 'language instinct' debate* (revised edition, 1999). London: Cassell.

Sommerville, I. 1992 *Software engineering*, 4th edn. Wokingham: Addison-Wesley.

Tesnière, L. 1959 *Eléments de syntaxe structurale*. Paris: Klincksieck.

Weinberg, G. 1971 *The psychology of computer programming*. New York: Van Nostrand Reinhold.

Discussion

Y. A. Wilks (*University of Sheffield, UK*). Can you clarify your analogy between NLP and software engineering, as the latter applies to well-defined problems where input–output specifications are clear and agreed, though stochastic techniques have been proposed as a means of generating such specifications.

G. Sampson. The analogy is intended to be looser and I do not want to suggest that any particular software-engineering methodology is relevant to NLP.

N. Ostler (*Linguacubun Ltd, Bath, UK*). Software engineering is neither glamorous nor a method for solving problems, but rather a way of ameliorating complexity. Would it not be better to try to solve the problems of NLP, as opposed to imposing a Linnean/Procrustean scheme on them?

G. Sampson. I feel that NLP still sees more glamour in coding and system development than in problem specification and documentation.

Downloaded from rsta.royalsocietypublishing.org

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A MATHEMATICAL, PHYSICAL & ENGINEERING SCIENCES

H. Cunningham (*University of Sheffield, UK*). Is there a wider role for taxonomy in uncovering redundancy and repetition in components of NLP systems? Might this not lead to more shared NLP tools, analogous to Mathematica?

G. Sampson. I doubt that the field is big enough to support development of a tool like Mathematica, but I agree that more publicly disseminated problem specification might lead to less duplication of effort.

S. J. Young (*University of Cambridge, UK*). Does NLP not need good science rather than software engineering? Unlike automatic speech recognition, the output specification is not generally agreed.

G. Sampson. I did not intend to discriminate between applied and theoretical work. Both require a shared problem description to make progress.

E. Hajíčová (*Charles University, Prague, Czech Republic*). Why do corpus annotation schemes differ? What are the operational criteria for assigning classifications?

G. Sampson. Different groups use different annotation schemes because there has been little public debate about such schemes as yet. Although well-defined operational criteria would be useful, development of such criteria tends to be in conflict with the requirement for comprehensive coverage of natural language data.

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A MATHEMATICAL, PHYSICAL & ENGINEERING SCIENCES