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

Zipf's Law and *L'Arbitraire du Signe*

Martin Kay

Published by CSLI Publications

LiLT volume 6, issue 8 October 2011

Zipf's Law and *L'Arbitraire du Signe*

Martin Kay

In every field of scientific enquiry, there is much data and therefore frequent cause to turn to the computer to help process it. This is certainly true of linguists. They use computers to search for examples of grammatical phenomena in large corpora and to collect statistics on their occurrence. They can use them to compile lexica, and to compare them with a view to assessing the relatedness of pairs of languages. Activities like these are collectively referred to as Natural Language Processing (NLP). Generally speaking, however, NLP is an engineering, rather than a scientific enterprise, much of it devoted to developing technologies, like machine translation, information retrieval, and speech recognition. It would be natural to expect these technological developments to be informed by the results of scientific enquiry carried out by linguists. In other words, it would be natural that they should have a foundation in computational linguistics. But this is rarely the case. Technological development in NLP is based almost entirely on machine-learning models most of which are wild and fantastical from a linguist's perspective. This, of course, is an aberration which, fortunately, may be in the course of correction.

In a tightly argued and largely convincing essay elsewhere in this volume, Steven Abney expresses a different view. "Computational linguistics", he writes, "is not a specialization of linguistics at all, at least not if we take "linguistics" and "computational linguistics" as academic communities defined by their membership." An academic community is a set of people and a set is surely defined by its membership, but

LiLT Volume 6, Issue 8, October 2011.
Zipf's Law and L'Arbitraire du Signe.

sets do not confer on their members the right to appropriate names already long since claimed by the members of other sets. In this paper, I shall continue to use the term "Computational Linguistics" to refer to an approach to the subject of linguistics that is informed and inspired by computing.

With Abney, I shall argue in this paper that "Language is a computational system, and there is a depth of understanding that is simply unachievable without a thorough knowledge of computation." There is a natural affinity between linguistics and computer science, and it is one that has very little to do with NLP. It arises because human language is one of very few naturally occurring phenomena that is fundamentally *digital*. Linguists and lay people alike tacitly acknowledge this affinity when they discuss such questions as whether spider is an insect, whether the vowel in "marry" is the some as the one in "merry", or whether I can claim simultaneously that "I heard about the argument in the library" while denying the truth of both "I was in the library" and "The argument was in the library". Notice that, while a spider may be more or less *like* an insect, it cannot be more or less an insect. Either it is, or it is not. Likewise with the vowels in "marry" and "merry". They may sound more or less different in the speech of different people, but the vowels of a particular English speaker's language constitute a small, fixed set and, in a given dialect, the vowels in these words are instances either of the same, or different members of that set. The sentence about the argument and the library has (at least) two syntactic structures, one of which puts me, and one which puts the argument, in the library. Language places the phenomena in its purview into absolutely discrete classes, and this is what makes it a digital system.

1 Introduction

Human language makes contact with the world at two places. It is used to talk about things in the world like objects, abstractions, thoughts, beliefs, facts, and fictions. This is one point of contact. Language also makes itself available to the senses through sounds, symbols, articulatory gestures, and marks on paper, which are also part of the world but which have no necessary connection to the first set of things. This is the other point of contact. The two are connected by processes that are essentially and crucially digital in nature and which therefore lie within the purview of computer science. The processes happen in peoples' heads. They are therefore hidden from view so that whatever we come to know about them must be based on the inputs and outputs to the process as a whole, that is, on the two places where they make

contact with the world.

Linguistics is concerned with every aspect of language and languages: how they came into being; how, over time, one changes into another; how children acquire them; how they serve to facilitate communication, and so on. The focus of attention shifts from one aspect to another over time. In nineteenth century Europe, the impetus came from the surprising observation by Sir William Jones (Jones, 1786) that Sanskrit was so similar to Greek, Latin, and other languages of the family we now call Indo-European, that they almost certainly all sprang from a common source. In twentieth century America, it came from the surprising observation by Franz Boas (Boas, 1911) and others that the native languages of America were so different from Indo-European, and from one another, that their study would require entirely new methods and assumptions.

Following the Swiss linguist Ferdinand de Saussure (de Saussure, 1915), the Americans embraced *structuralism*, insisting that the most important facts about a language concern the relationships that elements of the language—sounds, words, affixes, phrases—contract with one another, rather than with things in the outside world. They would concentrate on just one of the two points of contact. For de Saussure, structuralism was a theoretical stance. For the early American linguists, it was probably largely pragmatic for, in order to draw a coherent picture of the new and wildly different languages that they encountered, they were forced to examine much more closely than ever before fundamental issues such as the nature of the phoneme and the morpheme, and how to elicit reliable information from informants.

The direction changed again in the middle of the twentieth century with the publication of *Syntactic Structures* (Chomsky, 1957) which drew attention to what Chomsky saw as a more challenging problem, either than that of the genetic relationships that exist among some languages or of how one should describe languages that seem quite unrelated to any so far encountered. For him, the pressing problem concerned the *productivity* of language, that is, of how the speakers of any given language can produce and understand indefinitely many phrases and sentences in that language, presumably without having encountered them before. This *recursive* property is shared by all human languages, regardless of whether they are related in any other way. Since the concern was with what speakers know, and not only with what they do, it had to countenance mental entities and this required an abrupt change in the thinking of the linguists who decided to go in this new direction.

2 Language Data

Many linguists share a measure of insecurity about the true status of their discipline as a science, an insecurity having to do partly with the nature of the data that the discipline rests on, and partly with the fact that linguistics has had difficulty fully embracing the experimental methods that characterize other sciences. We started from the observation that language makes contact with reality at two places. Ferdinand de Saussure focussed on this fact, noting that language consists of a set of *signs*, each of which has two faces. To each word and phrase, to each *signifiant*, there corresponds a *signifié* which, at the coarsest level of granularity, is what it means or refers to. The first of these is more or less easily identifiable in the stream of speech or in written text, and constitutes the data that structuralists attend to. The second could include objects of any kind, real or imaginary, concrete or abstract, constant or evanescent. Furthermore, this duality is to be found not only on the level of meaning and reference, but it is also, as we have already suggested, what distinguishes phonetics from phonology. The structuralists decision to focus only on the *signifiant* could only be a local and a temporary stratagem for it is clearly only the bipartite nature of language that enables it to function as a means of communication. The key observation is that the relationship between a *signifiant* and the corresponding *signifié* is entirely arbitrary. This is what he referred to as *l'arbitraire du signe*. It implies, for example, nothing of the meaning of a text in an unknown language can ever be discovered entirely on the basis of internal evidence.

In some sciences, like physics and chemistry, data is obtained mainly from experiments—by perturbing the world in some way and observing how it reacts. This is generally regarded as the best way. When this is not possible, as in astronomy, the data comes from observing how the world reacts to whatever stimuli it happens to receive for whatever reason. The structuralists were sensitive to the many dangers of trying to apply the experimental method, especially to languages of which they had absolutely no prior knowledge. However, they provided a minimal paradigm for experimentation in linguistics that has been of crucial importance to linguists of every persuasion, based on the use of *minimal pairs*. This is the classical paradigm of the controlled experiment, transposed into the situation in which a linguist is eliciting information from an informant. Let us say that the linguist hypothesizes that the distinction his British informant makes between the vowels in *cart* and *gone* is absent in the speech of his American informant. If he asks his informants about these two words, he can never be sure whether

the judgements he gets are indeed based on their vowels or on some other properties that they have. He therefore looks for a different pair of words in which the vowel constitutes the only difference, say, *bomb* and *balm*. These constitute a minimal pair. If the British informant can always tell them apart when he hears them, but the American informant cannot, then his hypothesis is confirmed. If the American always distinguishes them accurately, then it can only be on the basis of the vowels which must, therefore, be different.

A linguist with any experience of field work knows that working with informants has many pitfalls, even for an experienced investigator. Some of these have to do with poor technique on the part of the linguist or inattention on the part of the informant, and these do not concern us. One problem has to do with the fact that the informant may believe that the members of his community always say one thing, and never some other thing, in a given circumstance, even though this is not the case. My father was sure that he always said "This needs to be done", and never "This needs doing" until I had caught him in the act several times. He was even more surprised to learn that he occasionally said "This wants doing". This effect doubtless comes from partial assimilation of prescriptive grammar in schools, and may or may not be present in the prototypical naive informant.

Another problem that arises when working with informants is that of incorrectly supposing that the language always requires p in a context q because either p or q has not been sufficiently well specified. I may conclude that the negation of a simple intransitive sentence like (1a) involves introducing the auxiliary *do* as in (1b) except when the sentence already contains an auxiliary, as in (1c), in which case *do* is not required, so we have (1d).

(1) a. Kim sleeps.
    b. Kim does not sleep.
    c. Kim will sleep.
    d. Kim will not sleep.

The linguist, and his colleagues, may hold to the hypothesis for a long time, before one of them encounters an example like (2)

(2) a. Kim dares to sleep.
    b. Kim dare not sleep.

These problems, as well as numbers of others involved in getting reliable and comprehensive information from informants, can be mitigated to some extent with computers and suitable corpora of texts in the language of interest. A search for the string "does not sleep" in a

large corpus of English would doubtless return a significant number of examples. A search for "does" and "sleep" with one intervening word should turn up a large number and a quick check of the intervening words would reveal most of them to be verbs. A search for "sleeps not" might reveal some unexpected examples like "He sleeps not because he is tired but because night is the time for sleep" and "Is the fact that he sleeps not important?".

The structuralists acknowledged the study of the relationships that *signifiants* contracted among themselves could never be studied in total isolation from the corresponding *signifiés*. Bloomfield wrote:

> ... phonology and, with it, all the semantic phase of language study, rests upon an assumption, the fundamental assumption of linguistics: we must assume the *in every speech community some utterances are alike in form and meaning* (Bloomfield, 1933, p. 78).

However, it was generally thought prudent to appeal to the *signifié* only for the purpose of helping to determine whether two signs were the same or different. In computational linguistics and NLP, the question of whether things are the same or different can be deferred for a while because the data usually consists of written text where the question of what is a character has been solved by *fiat* in advance. The same is true, at least in many languages, of a word. But if these are the only linguistic entities that are recognized, little of real interest can be achieved.

3 Digital Systems

A system in which objects from two domains are associated in arbitrary pairs is very likely to be either one in which at least one of the domains is digital. The sounds of speech live in the analog world and the more closely one examines them, the more differences one finds among them. Phonemes live in a digital space and two short segments of speech correspond either to the same or to different phonemes. A given dialect contains a certain number of phonemes, usually more than ten and less than a hundred. The phonemes in *bomb* and *balm* are absolutely different in most British, and absolutely the same, in most American speech.

Processing that is internal to the linguistic system is entirely digital, and crucially bipartite. Consider the lexicon. English speakers have different interpretations of the word "lay" in the sentences in (3). For those that say (3a), (3b) is in the past tense. Those for which (3b) is present tense speak a dialect from which (3a) is absent, or involves and altogether different verb.

(3) a. They lie on the couch.

b. They lay on the couch.

But there is no dialect in which "lay" is tenseless, or in which it has a tense that is intermediate between present and past, because tense is part of the digital system of all English dialects and every finite verb, in a given context, belongs to just one of the two available tenses. In phonology, morphology, and syntax, the digital nature of language is easy to illustrate. It is just as strong in semantics, but here there is also an analog component so that, just as one must be careful to distinguish analogue phonetics from digital phonology at the lowest linguistic level, so at the highest, one must be careful to distinguish meanings from referents. In the dialect in which I was raised, *jam* and *marmalade* mean different and, indeed, incompatible things. They are similar in most respects, but marmalade is made with citrus fruit and jam is not. For many English speakers, this does not hold. For them, marmalade is simply a kind of jam. For everyone, the lines that separate the categories are sharp. These are not facts about the substances in the real world that we refer to as "jam" and "marmalade" but about a system used to categorize them. Presented with a jar containing a jam-like substance made from strawberries and oranges, I would be at a loss for a word by which to refer to it. My dilemma would be one of connecting the digital to the analog world and not within the digital world of language itself.

3.1 Processes

Language presumably evolved to facilitate communication among people. The first person encodes information taken from one side of de Saussure's arbitrary line and encodes it in a sequence of symbols. The second decodes the symbols, and the line is crossed in the other direction. Since the communication system is itself embedded in the real world, the symbols must be given some analog representation before they can be transmitted so that both the speaker and the hearer must carry out both encoding and decoding processes. The analogue-digital divide is crossed four times in all for each utterance.

Most of the processing apparently takes place in the digital realm where we cannot observe it directly. However, by any reasonable measure, it appears to be quite complex. Linguists generally think of it as taking place on a number of different levels like phonology, morphology, syntax, and semantics. We generally take it that the message has a representation on each of the levels. On some of them, those closer to phonology, the representation is more similar to the encoded

message, and at others, closer to semantics, it is more similar to the intended meaning. Much effort in linguistics goes into trying to discern the properties that these various representations should have and the rules according to which representations on one level are translated into representations on adjacent levels. We must resort to often very indirect methods to infer the details of these processes, often leading to much discussion and disagreement. But the basic claim, namely that communication in general, and communication using human language in particular, involves much processing of abstract entities that can stand in many and complex relations to one another is hardly controversial. In other words, the problem of how communication occurs using natural language is, at least in part, a classical problem for computer science. It takes someone with a linguist's training and experience to observe and categorize the data; it takes a computer scientist to design data structures and describe processes that might lie behind it.

Following Chomsky (1965), linguists have come to recognize an important distinction between *competence*, the knowledge that speakers and hearers have of their language, and *performance*, the ability to deploy that knowledge in communication and for other purposes. One might expect the discussion of competence to be conducted largely in static, or declarative, terms and that performance would turn out to be the realm of process. But the linguists whose tradition gave us this distinction and who have made the most out of it, employ a vocabulary in their discussions of competence involving process-oriented terms like *generation*, *derivation*, *transformation* and even *computation*. They make frequent use of diagrams looking for all the world like flow charts with boxes for components dedicated to particular kinds of operation.

Other linguistic traditions also arose during approximately the same time period which, while they recognized the importance of the distinction between competence and performance in varying degrees, wanted to be able to claim that the processes embodied in their competence models are close to those that actually take place in the heads of speakers and hearers. This second tradition was based more explicitly in computer science. It includes Lexical Functional Grammar (LFG) (Bresnan, 1982, Dalrymple, 2001), Head Driven Phrase Structure Grammar (HPSG) (Pollard and Sag, 1994), Tree Adjoining Grammar (Joshi et al., 1975), and Combinatory Categorial Grammar(CCG) (Steedman, 2000). These are all concerned mainly with syntax, which is usually acknowledged to be the most complex part of grammar, but there were also many proposals in phonology, such as optimality theory (Prince and Smolensky, 1993), various approaches based on finite-state transducers(e.g. Kaplan and Kay (1994)) and so on. Generally speaking, these

proposals were made with the hope of reflecting more closely the processes that actually occur in the heads of speakers and hearers.

A follower of the second tradition might argue somewhat as follows. Presumably, the most important linguistic processes are those involved in the encoding and decoding of messages, that is, in the generation and analysis of utterances. These are both done in a accordance with the rules of a grammar and there is not a separate grammar for each of them. To know a language is to know its grammar and how to apply that grammar both in generation and analysis. The grammar must therefore have a sufficiently abstract form to be used with equal facility for both purposes including, of course, any processes that are part of it.

The beginnings of the other tradition in 1957 is strongly associated in the memories of those old enough to remember with the *passive transformation*, a rule which took as input a labeled, ordered, tree structure characteristic of an active sentence with a subject, a transitive verb, and an object. It delivered as output another labeled, ordered, tree structure characteristic of a passive sentence. The subject of the original had been moved to become the object of the preposition *by*, and the original object was moved into the position of the subject. The original verb now had the form of a past participle and its original tense became the tense of the auxiliary verb *be*.

A grammar would contain a *base component* consisting of context-free rules. These would generated *base structures*, including those of active sentences, to which an ordered list of transformations, some obligatory and some optional, would be applied to produce surface-structure trees, including those of passive sentences.

Shortly after transformational rules were first proposed, computational linguists that were sympathetic to this approach began looking for ways of using them in the automatic analysis of sentences (Kay, 1967, Petrick, 1973, Zwicky et al., 1965). This proved to be a formidable task for a variety of reasons. The most obvious way to approach it, and the one which was attempted several times with only minor variations, involved constructing a rule corresponding to each of the transformations that would reverse the effect of the original. Like the original rule, this reverse rule carries a tree onto a tree, so that there would, for example, be one carrying a characteristic passive sentence onto its active counterpart.

The question of how to use such rules leads to a number of serious problems. The first is that of constructing the first tree in this new sequence, that is, the tree that would be the last in the sequence followed by the original grammar in generating the sentence. The analysis process would presumably have to take this as input, but the input that is

supplied is, of course, only the sequence of terminal symbols from this tree.

Another problem comes from the fact that rules that are to be carried out obligatorily in the forward direction do not, in general, correspond to obligatory rules in the reverse direction because one and the same tree could easily result from quite different sequences of rules. One need only consider a sentence like "the old man was found by the church" which, in its preferred interpretation is clearly not passive. In parsing it, the passive rule would therefore have to be treated as optional.

A third problem, which was addressed in later versions of the theory, had to do with the fact that transformations could delete arbitrary material which would have to be reconstituted in some manner by the reversed process. As a result of these various considerations, the parsing grammar generally required more transformational rules than the original grammar contained.[1]

To get the first tree in the reversed process, a special surface context-free grammar was constructed. No way of making this part of the process completely automatic was ever devised. A linguist was therefore called upon to write a set of rules that would enable surface structures to be constructed over the strings to be analyzed, and the reversed transformational rules would be applied to these to construct putative base structures.[2] This surface grammar was required to generate all the structures that the transformational grammar could generate, though it might also generate others, which would presumably be eliminated when the reversed rules were applied. Owing mainly to the fact that reversed rules could not be presumed to be obligatory, structures derived by the reversed rules were generally, at best, a superset of the correct ones, so that it was necessary to check that each of them would in fact generate the original sentence. In other words, the whole original transformational grammar had to be run on each proposed base structure to be sure that the given sentence could indeed be derived from it. No satisfactory solution to these problems was ever found, and continual changes in the underlying theory provided computational linguists with a target whose movements they had little chance of tracking. This line of work was therefore abandoned for a long time and has only recently started to attract attention again(e.g. Stabler (2001)).

Partly in response to the situation just described, early computational linguists tried to design formalisms of their own that could achieve similar results but would prove more tractable. They began

[1]The MITRE grammar contained 54 forward, and 134 reversed transformations

[2]The MITRE grammar had base grammar of 275, and a surface grammar of 550 context-free rules.

by devising formalisms designed to favor the hearer rather than the speaker, that is, parsing rather than generation. Perhaps the most interesting, and certainly one of the most successful of these proposals was that of *Augmented Transition Network(ATN's)* (Woods, 1973) which, in its turn, were based on *Recursive Transition Networks (RTN's)*. A network is little more than a directed graph with labeled edges so that it is similar in most ways to a finite-state machine. The distinguishing characteristic lies in how the labels on the edges are interpreted.

An RTN consists of a set of networks, differing from finite-state machines in that a transition from one state to another could be made either if its label matched the next input symbol or if the label named another network which accepted a subset of the input beginning at the current position. A sentence network could thus contain a noun-phrase network that would match a more or less complex subject, after which work in the sentence network would resume to match the verb and the rest of the sentence. The noun-phrase network that was used to identify the subject would usually be called upon again to analyze the object.[3]

RTN's have the same formal power as context-free grammars, but are able to generate flatter structures. The idea behind ATN's was to enrich the capability of RTN's so that they would acquire power similar to that of transformational grammars, which, in most of their incarnations, was that of Turing machines. To the extent that this was achieved, it was mainly through the introduction into RTN's of *registers* which were very much like the variables of most programming languages. Arbitrary values could be stored in registers and copied from one register to another. The path taken through the network could be dependent on the content of registers. The structure assigned to a substring of the string under analysis was assembled from the contents of various registers so that it did not have to reflect the pattern of recursive calls made by one network to another.

Translating the formal notation into English, a toy grammar for simple sentences might look somewhat as follows:

1. Call the NP network. Place the result in the *subject* register. Go to node 2.
2. If the current item is a verb, put it in the *verb* register and its tense in the *tense* register. Go to node 3.
3. If the current item is the past participle of a transitive verb and the *verb* register contains a part of the verb *to be*, put the current item in the *verb* register, and go to 5; otherwise go to node 4.

[3]For present purposes, we gloss over details, such as agreement and the differential treatment of pronouns.

4. Call the NP network. Place the result in the *object* register. Go to node 6.
5. Put the content of the *subject* register into the *object* register. Call the NP network and put the result in the *subject* register. Go to node 6.
6. Return the structure [category: sentence, subject: *subject*, verb:*verb*, tense: *tense*, object:*object*] (Italicized words are the names of registers.)

ATN's have been among the most successful kinds of grammar for parsing, but they clearly have the obverse of the problem that transformational grammars have, namely that they cannot, in general, be reversed and used for generation. Notice that, in the toy grammar given above, the contents placed in the *subject* register at node 1, are replaced in node 6. When attempting to reverse the process, we must therefore try placing the first NP encountered in a right-to-left scan of the sentence in both the *subject* and the *object* registers independently. In the end, we must verify any putative result by parsing the resulting string. In a grammar of realistic size, the extent of the resulting computations can easily become overwhelming.

The attempt to make ATN's reversible led directly to a proposal that proved to be pivotal in the search for more abstract grammars. The proposal was to eliminate variables of the kind encountered in everyday programming in favor of variables used in mathematics and logic, and replace equality tests by *unification* in matching operations. This move, in due course, let to the development of formalisms like LFG and HPSG. What was important about these formalisms, from the point of view of the present discussion, is not so much the extent to which they accommodated the processes required for generation and parsing, as the level of generality at which these processes could now be characterized. Unlike either grammatical transformations or ATN's, they do not depend on a minutely specified sequences of events leading from one state to the next. Instead, they rely on relatively simple algorithms, like those involved in unification (Kay, 1984) and chart parsing, whose minute details could be implemented in a variety of ways because their details are not crucial to the success of the enterprise. These developments resulted from the close cooperation between theoretical linguists and computer scientists that gave rise to computational linguistics, a development could doubtless not have arisen in any other way.

One may object that neither transformational grammarians nor computational linguists can make a credible case for the psychological plausibility of their models. Grammatical transformations, and the pro-

cesses that have replaced them in more recent versions of the theory, operate on complete sentence structures rather than generating them from left to right as humans apparently do. Chart parsers minimize the complexity of the process through dynamic programming, which requires one to assume that the early part of a sentence is still remembered in complete detail when the later parts are being processed. However, to a large extent, the computationally based theories separate the specification of the process from the specification of grammatical constraints so as to allow considerable variation in one without greatly affecting the other. A chart parser in which new edges are placed on an agenda from which they are removed in a priority order to be put in the chart allows the aspects of the process that are needed to assure a correct final outcome are largely separated from those that would be required to model a particular psychological strategy.

This last point is important for the present line of argument. Linguistic theory should indeed aim to characterize the processes involved in sentence generation and analysis, but to attempt this at a level of detail that involves specifying actual sequences of operations will not only prejudice the theory in favor of either the speaker or the hearer, but will obscure any general claims that the theory makes. We should look to the natural symbiosis between linguistics and computer science to give rise to more general and more robust ways of characterizing these processes. Agenda-driven charts and parsing (Kay, 1986) are a step in this direction.

As we remarked at the outset, a field of endeavor that is so closely related Computational Linguistics as to be confused with it in the minds of many is referred to as *Natural Language Processing (NLP)*. The first is the scientific enterprise we have been discussing which addresses the same questions as the larger field of linguistics, but from a computational perspective. The second is an engineering enterprise addressing practical problems that involve language in non-trivial ways. Computers are, of course, used in crucial ways by scientists in many fields who do not, however, feel impelled to distinguish themselves as "computational" physicists, geneticists, or whatever. What I take to be important about computational linguists is not so much that they use computers as that they deliberately and self-consciously seek inspiration not so much in computers themselves as in computer science.

It should come as no surprise that Computational Linguistics falls together with NLP in the minds of many people because the motivation for much of the work that has been done in computational linguistics came originally from engineering concerns. The very term "Computational Linguistics" was coined in response to the 1966 report the Auto-

matic Language Processing Advisory Committee to the US government (John R. Pierce, 1966), urging more fundamental scientific investigation before any further work be undertaken on machine translation. What I have just tried to argue is that the scientific effort that the committee called for was indeed undertaken and has proved remarkably successful so far. As we have also noted, linguists of all kinds have also used computers in more pedestrian ways, to search corpora for examples and counterexamples and, while this generally has little to do with computational linguistics, it does contribute to the confusion.

4 Machine Learning

Two other fairly recent and related developments confuse matters a great deal more, and should be a cause for concern among linguists and practitioners of natural language processing. One is the dominant status that machine learning has acquired in NLP. The other is the appeal that linguists are beginning to make to statistics as a basis on which to explain linguistic facts.

Language processing of almost all kinds generally involves dictionaries from which information about words and phrases can be retrieved, and rules that can be applied to some or all of a text in order to translate it, retrieve information from it, or whatever the task happens to be. Writing dictionary entries and rules is tedious and therefore expensive, subjective and therefore error prone. The idea behind machine learning in this context, is to replace the human lexicographer and rule writer by a computer program that will have acquired the necessary skill by assimilating the linguistic information in huge quantities of naturally occurring text. The machine does not approach the task with preconceived notions about how things should be. It does not tire or get bored. It works fast and will be happy to do the job over again if the requirements change, even very slightly.

Machine learning takes many forms, but it generally fits one of two paradigms, known as *supervised* and *unsupervised* learning. Learning is said to be supervised when it is based on some number of correct solutions to the kind of problem it will be called upon to solve. Part-of-speech taggers generally learn from texts that have been tagged by humans, presumable correctly. Such taggers therefore are products of supervised learning. A morphological analyzer could be built in the same way, training on examples of correct morphological analyses, and it would therefore be said to result from supervised learning. But a morphological analyzer might also be built on the basis of raw text by considering the frequencies of letters or sounds at the beginning

and ends of words, and it would therefore result from unsupervised learning. Statistical machine translation systems are the product of supervised machine learning in that the data on the basis of which they are constructed consists of large numbers of sentences, each with a presumably correct translation. On the other hand, pairs of texts, one of which is a translation of the other, occur naturally, so that they do not have to be constructed especially for the purposes of learning. From this point of view, the standard way of training statistical machine translation systems looks more like unsupervised learning.

There is very little to be learned about natural language by techniques that are clear cases of unsupervised learning because of the arbitrary nature of linguistic signs to which de Saussure drew attention. The meaning of a text, and whatever it refers to in a real of imaginary world, is accessible only to someone that knows the language. Jean-François Champollion would have been quite unable to decipher Egyptian hieroglyphs if it had not been that the Rosetta also had a parallel text in Greek, a language which he knew. He was thus able to achieve in a very small way what the producers of modern statistically based machine translation schemes achieve on a massive scale.

Machine translation is the archetypal task for both computational linguists and practitioners of natural language processing. It is a historical fact, as I have already remarked, that it played a key role in the birth of both fields. It has generally been acknowledged as the most demanding task that one could undertake in both fields because it seems to require solutions to just about all the problems of linguistic processing that one can imagine, both scientific and engineering. Indeed, it requires a great deal more because, the longer one looks, the more apparent it becomes that the problems that translation pose extend far beyond the realm of language to cover most of artificial intelligence in addition. The justification for this claim has been set out in considerable detail elsewhere (Kay, 1997). The basic point is that there is much in any but the most trivial texts that a reader must infer from what is made explicit, but what is explicit in a translation in another language is not generally the same, so that substantive information is both added and subtracted by the translator. This is not something that happens as a result of carelessness or bravado on the part of the translator; it is something that is necessary to accommodate one language and culture to the other.

Consideration of just how how much we should expect of machine translation systems that learn from examples of translation, however extensive and however expertly done, raises a very serious philosophical question that is, however, rarely addressed. The problem is simply

this: just how much of what translators know that enables them to do their job is captured in their products so that it could, in principal, be extracted, even in a probabilistic form, from them? Could one say that the extractable information approaches all one could need, in the limit, if unbounded amounts of material were considered? The evidence strongly suggests that the answer is "no". A few examples should suffice to make the point.

It has become standard practice to train statistical machine-translation systems on a corpus consisting of millions of words taken from the proceedings of the European Parliament. The following examples come from a two-minute examination of first 300 words.

English the dreaded - 'millennium bug'

French le grand 'bogue de l'an 2000'

If the French were the original, then it is hard to see how "grand" could have given rise to "dreaded" in English except, of course, in the mind of someone who knew what the millennium bug was, and how it was perceived by the public.

English I should like to observe a minute's silence

French je souhaiterais ... que nous observions une minute de silence

I am inclined to think that the original was in English, and that a pedant would say that the speaker had make a mistake. The French translator corrected the error, suggesting that all those present should observe a minute's silence, and not just the speaker. Translators are kind and helpful and they do this kind of thing all the time.

English The House rose and observed a minute's silence

French Le Parlement, debout, observe une minute de silence

To translate the English sentence into this French one, you need to know that, after a person has "risen" they are "debout". On the other hand, to translate this French sentence into the given English one, you need to know that, while a person normally "stands up" or "gets up", when they are about to observe a ritual moment of silence, they "rise". More striking is that the English "rose" is a verb, whereas "debout" is an adjective. A system that could surely learn to translated these sentences in this way only at the expense of translating many others disastrously.

English Would it be appropriate for you, Madam President, to write a letter ...

French Ne pensez-vous pas , Madame la Présidente , qu'il conviendrait d'écrire une lettre ...

The English talks about a letter being written by the president, but the French talks about the appropriateness of writing a letter, but makes no mention of who should write it. Furthermore, the French says, in effect, "do you not think that it would be appropriate to write a letter", but the English says nothing about what anyone does, or does not, think

Situations in which one must know a great deal more than the vocabulary and grammar of the languages involved in order to do what translators do are endless. In at least some of the above cases, a different translation could have been given which would have required less nonlinguistic knowledge, but many remain for which this is not possible and, in any case, it is surely legitimate to ask why translators so often prefer these less "strict" translations.

Since examples of the kind just considered are clearly beyond the reach of current, or any readily foreseeable technology—especially if based on machine learning—we must take it that they do nothing but degrade the best performance of the systems that are learned from the texts that contain them. Supervised learning from a corpus of translations that were stricter, if less idiomatic, should surely be expected to result in superior systems. But large corpora of such translations do not occur naturally, would be expensive to produce artificially, and would be required to meet different criteria as the field progressed. One might however approximate it in the following way. Having trained a system on a given corpus, remove from that corpus some proportion of the sentences that it translates badly in the hope that these would include many of the difficult cases we have been considering. Then retrain the system on the remaining sentences.

The attempt to apply machine learning, and statistical techniques more generally, to natural language has, of course, not gone unnoticed by linguists themselves, some of whom have seen it as an opportunity to broaden their own horizons. The results are sometimes extraordinary. Bresnan (2007) asks the question "Is syntactic knowledge probabilistic?" On the face of it, this is a remarkable question in that it invites one to at least contemplate a world in which the answer would be "no". Presumably the world contains very little that is not, in one way or another, probabilistic. The intended sense is made clear in the following passage from the introduction to the paper:

> Moreover, studies of usage as well as intuitive judgments have shown that linguistic intuitions of grammaticality are deeply flawed, because (1) they seriously underestimate the space of grammatical possibility by ignoring the effects of multiple conflicting formal, semantic, and contextual constraints, and (2) *they may reflect probability instead of*

grammaticality.[4]

We are invited to place probability and grammaticality on an equal footing as alternative sources for linguistic intuitions as though probability were a source of causation in its own right. This seems to be an alarmingly strong version of the probabilistic hypothesis, relying, as it apparently does, on the existence of a real dice-throwing mechanism in the heads of speakers. This would place human language together with quantum mechanics as the only places in the universe where such a mechanism is contemplated. Pending more evidence on this remarkable view, it seems prudent to appeal to probability and statistics as means for managing uncertainty rather than as pillars on which scientific explanation rests.

The lesson is clear and, through most of history has really been in any doubt. The linguistics system as a whole, and every every example of its use has two equally important faces, one which is observed in the utterance or the text, and the other which exists in the heads of the sender and the receiver. Each of these is but a pale reflection of the other, a fact on which the enormous power of the system clearly rests.

5 Zipf's Law

Unsupervised learning seems to fit best the structuralist program for linguistics where it is patterns of purely linguistic objects that are of primary interest. So long as structuralism remained the dominant paradigm in linguistics, what a linguist most needed from an informant was judgements about the membership of the set of strings that made up the language. Since the concern was for the relationships that linguistic segments contracted with other members of this set, and not with anything outside, it was sufficient to determine which strings should have asterisks placed against them, thus declaring them not to be part of the language. Though the paradigm has changed, the assignment of asterisks has remained a favorite activity among linguists and a powerful tool in all kinds of linguistic research. A large corpus, a computer, and some suitable software for searching, might be expected to go a long way towards answering the question without any help from an informant at all.

We suppose that the utterances in the corpus were all found in reasonably unconstrained circumstances so that everyone is happy to say that they *occurred*. None of them gets an asterisk. Some words occur very frequently, as do some short utterances. Most occur only once. Even a structuralist would be unsatisfied if left with nothing to say

[4] italics mine

about long sentences or rarely occurring words. If some frequently occurring words are found only in a restricted set of environments where some infrequent words are also found, the latter may be taken to share grammatical properties with the former. To the extent that long and infrequently occurring utterances share substrings with shorter, more frequently occurring utterances, it may be possible to formulate some plausible hypotheses about their structure. But these will remain tentative at best because the measure of a structural analysis of an utterance is the support it provides for other analyses, particularly on the level of semantics, there information from the other side of de Saussure's divide is indispensible.

A corpus almost always contains many phenomena in such weak dilution that they do reveal nothing of interest to an investigator. His first impulse is therefore to go in search of more data. However, it seems that new data generally opens at least as many questions as it settles. The amount one can expect to learn about the system that underlies a language falls off very rapidly as one considers more text in the language. This is in accordance with Zipf's law (Zipf, 1935) which says that the frequency of a word is generally much greater than that of the next most frequent word, regardless of the length of the text. In particular, if f_0 is the frequency of the commonest word, the frequency of the n-th most common word, f_n, is about f_0/n^θ, where θ is close to one.

It is usual to exhibit the Zipf distribution with a graph in which the words are arranged on the x-axis in decreasing frequency order, with the frequencies themselves on the y-axis. Figure 1 shows the curve for Shakespear's *Hamlet*, a text of 4686 words. The most frequent word occurs 1101 times. 3827 words occur only once.

What is true of words is also true of sequences, phrases, grammatical categories, rules applied, and so forth. There is a small part of what we know about our language that we use constantly and a very large part that we use only very rarely. New data contains many new utterances but they are almost all constructed using words and grammatical devices that have already been seen before. What is new about them is what they say, not how they say it, and this is surely fortunate because it allows us to continue learning new things from the linguistic input we receive without having to learn corresponding numbers of new words and constructions.

As I have already suggested, the structuralist program is a fundamentally unsatisfactory one, and in the extreme case where text replaces the informant, it is especially so. What is unsatisfactory is that the program remains firmly on one side of the Saussurian divide. If

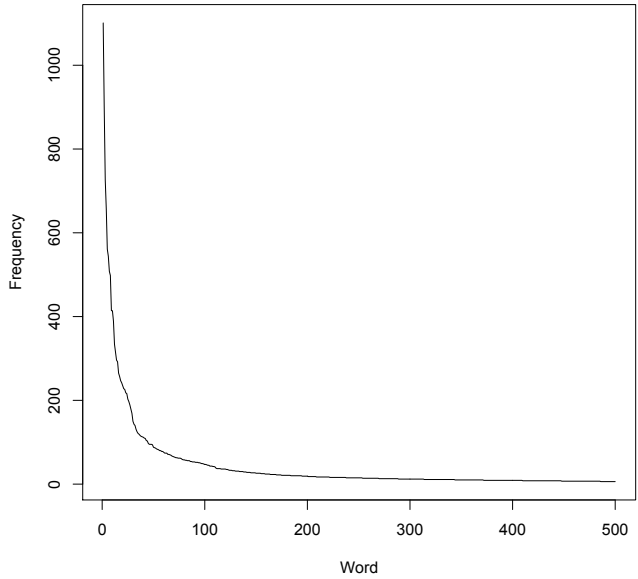

FIGURE 1 The Zipf Distribution

there is a computer and a corpus, then a representation of the corpus in terms of sequences of phonemes or characters must have been settled upon in advance. Beyond this, we have no information whatsoever about how the material in the corpus might figure in acts of communication. If we are interested in information retrieval, sentiment analysis, or the like, we have not a leg to stand on.

We have seen how workers in Natural Language Processing attempt to solve the problem. Realizing that text is fundamentally impenetrable to anyone who does not have the code, they commonly resort to a mixed approach based on supervised learning. They have speakers of the language annotate the corpus so that information about its words, phrases and utterances that are normally hidden are made explicit. They give each word a part of speech, each named entity a suitable tag, and so forth. The idea is that these are tasks that are close enough to what a person would normally have to do to understand the text in any case that they can do them easily and reliably. Furthermore, occasional errors in such a large cauldron of material will have essentially no effect.

Fortunately for the annotators, they are not generally paid according to the expected marginal value of the individual annotations they make. If they were, then their pay would fall off at an alarming rate. As the process continues, the incidence of new linguistically relevant phenomena annotated becomes extremely low, and it continues to fall until there is almost no return on investment at all. Learning about the subject matter, however, remains substantially constant. But, as we have already noted, this property of language, which works so strongly against structuralist and partially structuralist methods in linguistics, works strongly in favor of the user of the language. It allows for the introduction of new features to the language, but at a rate that is sufficiently low that comprehension is rarely seriously impaired. The flow of information that the language encodes, and which the language user is interested in, is not subject to Zipf's law but is determined by the extent to which the sender of the message has anything new to say.

Why is it, then, that advocates of machine learning in computational linguistics and NLP in general, and statistical machine translation in particular, constantly insist that "there is no data like more data"? We have, in fact, already hinted at the answer to this question. Practical tasks that appear at first glance to be concerned only with natural language, invariably involve a great deal in addition. Machine translation is a paradigm example. Very generally speaking, a statistical machine translation system consists of two parts: a translation model and a language model. The translation model proposes sets of words and phrases that might, with certain probabilities, translate individual words and phrases in an original sentence. The language model selects one candidate from each of set and arranges them into a string that looks as nearly as possible like a sentence of the target language. What makes a string of words look like a sentence in a given language? The answer is that as many of its substrings as possible occur frequently as substrings of sentences in a large corpus of text belonging to that language.

Perhaps the main thing that is wrong with this scheme is the term "language model". As we have seen, the larger this is, the less it appears to model the language, and the more it models what people have been observed to say in that language. The less it is a model of the language, and the more it becomes a model of the world. And, the argument continues, this is exactly what is required if, as I have been arguing, knowledge of the world plays a central role in achieving good translations.

While it is generally acknowledged that statistical machine translation systems work better on subject matter that is similar to that in their training data, this line of argument is rarely adduced and the term

"language model" remains firmly in place. The reasons are not hard to see. Attempts to build models of even very limited parts of the world using all the inventiveness that artificial intelligence has been able to bring to bear, have met with only very limited success. The proposal that the n-tuples of words found in naturally occurring text can constitute such a model clearly cannot be taken seriously. Notice that a large language model would, and should, guarantee that the French "homme mord chien" would be translated into English as "dog bites man", rather than "man bites dog", which is what it really means.

6 Experimental Linguistics

We began by citing Abney on his view that the term "computational linguistics" has come to refer to what I have persistently referred to as "natural language processing". At this point, it is appropriate to take up a more substantive point that he makes in that paper, and a suggestion, never made quite explicit which, if pursued to its logical conclusion, would be disastrous.

Abney claims, quite rightly, that very little linguistics is good science. Such attempts as there have been to apply the experimental method have been half-hearted at best and dishonest at worst. Evaluation criteria have generally been invented after the outcomes are known and have often involved an array of special cases to shoe-horn the outcomes into previously adopted orthodoxies, masquerading as theories. But a reaction to this state of affairs has set in, albeit belatedly (Cowart, 1997, Sprouse, 2007). Several universities have established Syntax laboratories and teach courses in lab syntax. The Stanford course "aims to provide students with a systematic introduction to methods of handling syntactic data, including corpus work on ecologically natural data and controlled experimental paradigms". The most striking result of these developments is that the absolute nature of grammaticality judgements—whether asterisks might come in various shades of gray—has had to be faced head on (Gisbert Fanselow and Vogel, 2004, Keller, 2000).

The suggestion is that the new computational linguistics, that I have been calling "natural language processing", might actually turn out to be the new experimental linguistics. Competent statisticians who work on large collections of naturally occurring texts do understand the experimental method and the enterprise they are engaged in is clearly only in its infancy. But this way leads to perdition for the reasons I have given. *L'arbitraire do signe* assures that every interesting question about a text has a bipartite answer, only one part of which is to be

found in the text. There is doubtless more information in a text that comes with a translation in another language, but probably no that much more. Zipf's law assures that each utterance examined provides less information about that language than the one before.

I have attempted to argue that linguistics and computer science are natural bed fellows because language is a digital system and computer science is the science of digital systems. Many of the advances that have been made in linguistics during the last half century are a direct result of the collaboration. In the last fifteen or twenty years, however, an entirely new line of enquiry has been introduced involving language and computing, championed by engineers for whom the scientific concerns of theoretical and computational linguists are uninteresting, and their methods considered unhelpful. Fortunately, this chapter in the history of our field will be short lived. Already, the realization is growing that languages have morphologies, that adjacency in the string is the wrong domain of locality for many linguistic purposes, that sentences have recursive structures, that there is a difference between what you say and the language you say it in. For their part, linguists are coming to appreciate that, while statistics explain nothing by themselves, they can cast light on what needs to be explained and, perhaps, where to start looking for the explanation. In particular, they have cast doubt on the supposition that informants can be regarded as oracles, providing complete and reliable information on their languages.

References

Bloomfield, Leonard. 1933. *Language*. New York and London: Henry Holt and Co. and Allen and Unwin Ltd.

Boas, Franz. 1911. *Handbook of American Indian Languages*, vol. 1, chap. Introduction, pages 5–83. Washington D.C.: Bureau of American Ethnology.

Bresnan, Joan, ed. 1982. *The Mental Representation of Grammatical Relations*. Cambridge, MA: The MIT Press.

Bresnan, Joan. 2007. Is syntactic knowledge probabilistic? experiments with the English dative alternation. In S. Featherston and W. Sternefeld, eds., *Roots: Linguistics in Search of Its Evidential Base*, Studies in Generative Grammar, pages 77–96. Mouton de Gruyter, Berlin.

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

Chomsky, Noam. 1965. *Aspects of the theory of syntax*. Cambridge, Massachusetts: M.I.T. Press.

Cowart, Wayne. 1997. *Experimental syntax : applying objective methods to sentence judgments*. Thousand Oaks, Calif.: Sage Publications. 96035620 Wayne Cowart. Includes bibliographical references (p. 175-177) and indexes.

Dalrymple, Mary. 2001. *Lexical Functional Grammar*. No. 42 in Syntax and Semantics. New York: Academic Press.

de Saussure, Ferdinand. 1915. *Course in General Linguistics*. London: Peter Owen Ltd. W.Baskin (1959).

Gisbert Fanselow, Mathias Schlesewsky, Caroline Féry and Ralf Vogel. 2004. *Gradience in Grammar*. Oxford University Press.

John R. Pierce, et al., John B. Carroll. 1966. *Language and Machines — Computers in Translation and Linguistics*. Washington, D.C.: National Academy of Sciences, National Research Council.

Jones, Sir William. 1786. The third anniversary discourse, delivered 2d february, 1786: on the hindus. In *Asiatick Researches*, vol. 1, pages 415–431. Asiatick Society of Bengal.

Joshi, A. K., L. S. Levy, and M. Takahashi. 1975. Tree adjunct grammars. *Journal Computer Systems Science* 10(1).

Kaplan, Ronald M. and Martin Kay. 1994. Regular models of phonological rule systems. *Computational Linguistics* 20:331–378.

Kay, Martin. 1967. Experiments with a powerful parser. In *Proceedings of Conference Internationale Sur Le Traitement Automatique Des Langues (COLING 1967)*. Grenoble.

Kay, Martin. 1984. Functional unification grammar: A formalism for machine translation. In *Proceedings of the 10th International Conference on Computational Linguistics, Stanford, CA*. Association of Computational Linguistics.

Kay, Martin. 1986. Algorithm schemata and data structures in syntactic processing. In K. S. J. Brabara J. Grosz and B. L. Webber, eds., *Readings in Natural Language Processing*, pages 35–70. Los Altos, CA: Morgan Kaufmann.

Kay, Martin. 1997. The proper place of men and machines inlanguage translation. *Machine Translation* 12(1/2):3–23.

Keller, Frank. 2000. *Gradience in Grammar: Experimental and Computational Aspects of Degrees of Grammaticality*. Ph.D. thesis, University of Edinburgh.

Petrick, S. R. 1973. Transformational Analysis. In R. Rustin, ed., *Natural Language Processing*. New York: Algorithmics Press.

Pollard, Carl and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. Chicago, Illinois: University of Chicago Press and CSLI Publications.

Prince, Alan and Paul Smolensky. 1993. *Optimality Theory: Constraint Interaction in Generative Grammar*. MIT Press.

Sprouse, Jon. 2007. *A Program for Experimental Syntax*. Ph.D. thesis, University of Washington.

Stabler, Edward P. 2001. *Computational Minimalism: Acquiring and Parsing Languages With Movement*. Blackwell.

Steedman, Mark J. 2000. *The syntactic process*. Cambridge, Massachusetts: MIT Press.

Woods, W. 1973. An Experimental Parsing System for Transition Network Grammars. In R. Rustin, ed., *Natural Language Processing*. New York: Algorithmics Press.

Zipf, George K. 1935. *The Psychobiology of Language*. Boston, MA: Houghton-Mifflin.

Zwicky, Arnold M., Joyce Friedman, Barbara C. Hall, and Donald E. Walker. 1965. The MITRE syntactic analysis procedure for transformational grammars. In *Proceedings of the November 30–December 1, 1965, Fall Joint Computer Conference, part I*, AFIPS '65 (Fall, part I), pages 317–326. New York, NY, USA: ACM.