*L*inguistic *I*ssues in *L*anguage *T*echnology – *LiLT*
Volume -, Issue - DRAFT 2010

Data-Intensive Experimental Linguistics

Steven Abney

Published by CSLI Publications

Data-Intensive Experimental Linguistics

Steven Abney, *University of Michigan*

Judging by the name, one would expect computational linguistics to be a specialization of linguistics, and indeed the Association for Computational Linguistics defines it as "the scientific study of language from a computational perspective" [Sproat (2005)]. In a discussion of its genesis, Martin Kay emphasizes that "computational linguistics is trying to do what linguists do in a computational manner" [Kay (2005)].

But, in fact, computational linguistics is not a specialization of linguistics at all; it is a branch of computer science. A large majority of computational linguists have degrees in computer science and positions in computer science departments. We will examine the history of the field shortly, but in precis, it was founded as an offshoot of an engineering discipline (machine translation), and has been subsequently shaped by its place within artificial intelligence, and by a heavy influx of theory and method from speech recognition (another engineering discipline) and machine learning.

Nonetheless—and more by necessity than choice—computational linguistics has developed its own philosophy and methodology of language study, a body of practice that effectively constitutes an alternative linguistics: a linguistics characterized by systematic data collection and rigorous, experimental testing of predictions. Let me emphasize that its subject matter is language, not technology. It makes sophisticated use of computation, but no more so than do astronomy, biology, physics, or any other modern science that studies complex dynamic sys-

LiLT Volume -, Issue -, DRAFT 2010.
Data-Intensive Experimental Linguistics.

tems and relies on large-scale data collection. Computation is a means to an end; the essential feature is data collection, analysis, and prediction on the large scale. I will call it **data-intensive experimental linguistics**.

Most of what follows will be familiar, even elementary, to computational linguists. I address myself primarily to linguists. I wish to explain how data-intensive linguistics differs from mainstream practice, why I consider it to be genuine linguistics, and why I believe that it enables fundamental advances in our understanding of language.

1.1 A Brief History

In any discussion of the relationship between linguistics and computational linguistics, the elephant in the room is the enormous gulf between the fields. Given their history, as we shall see, that gulf is actually unsurprising. It has led to computational linguistics developing its own, alternative philosophy of linguistics. But on closer consideration, one can view that alternative approach as a reinvigoration of a mathematically sophisticated linguistics that blossomed—prematurely, it would seem in hindsight—fifty years ago. I will even suggest that it can be seen as a revival of Chomsky's original conception of generative grammar.

1.1.1 The making of computational linguistics

According to a first-hand account by Martin Kay (2005), the term *computational linguistics* was invented by David Hays in 1962. As Kay describes it, the term—and field—were created in response to what would eventually appear in the now-infamous ALPAC report (1966). The report evaluated the accomplishments and prospects of machine translation research; it would conclude that machine translation neither filled a genuine need nor was likely to be successful any time soon; and it would criticize machine translation as an engineering field whose long-term potential was hampered by a lack of deeper scientific foundations. Hays knew the report was coming—in fact, he would help write it—and, as a preemptive measure, he coined the term and founded the Association for Machine Translation and Computational Linguistics, which in 1968 became simply the Association for Computational Linguistics [Sparck Jones (2002)]. In this light, it seems that computational linguistics had little to do with linguistics, but was founded as a "scientific arm" of machine translation, at least in part for funding-political reasons.

An alternative name that Hays rejected as insufficiently scholarly was "natural language processing," a name that was later adopted by artificial intelligence researchers for the language-processing compo-

nents of an artificial intelligence. Beginning in the late 1960s, work on natural language processing in artificial intelligence (AI) constituted a second major tributary to computational linguistics, above and beyond machine translation. Natural-language access to databases became a popular vehicle for AI-oriented research, though the practical exigencies of building database front-ends often eclipsed the larger vision.

At the end of the 1980s, there were large intellectual influxes from two additional fields: speech recognition and machine learning. Through them, computational linguistics absorbed an enormous body of material from information theory, coding theory, statistics, pattern recognition, and operations research. As a result, the field has been transformed from a corner of applied computer science known mostly for system building, to a discipline that can compete with the natural sciences in mathematical sophistication.

Interactions between linguistics and computational linguistics have not been entirely lacking. Arguably the most productive interaction is represented by work within unification-based formalisms, whose practitioners have included both linguists and computational linguists. Unification-based formalisms were a major focus of research during the 1980s, and features and unification have become part of the standard curriculum in computational linguistics.

Even so, the unification-based work may be the exception that proves the rule. One of its central principles is a clean separation between the grammar-writer and the implementor. Ideally, the formalism permits linguists to write grammars without concern for the algorithms that will be applied to them, and it permits computational linguists to write algorithms without knowing the contents of the grammars, any more than the implementor of a Fortran compiler need be cognizant of astronomy or biology or any other subject domain for which Fortran programs are written.

To be sure, the ideal has never been entirely achieved. Nonetheless, instead of providing a conduit for mutual influence between linguistics and computational linguistics, the most successful interaction between the fields aimed for a strict division of labor between them.

Given this history, it should be no surprise that there is a gulf between the fields. In fact, any expectations to the contrary might now appear to be based on little more than Hays's personal sense of what sounded scholarly in a name.

1.1.2 A new philosophy of linguistics

And yet ... no field that works so closely with language can fail to "do linguistics" in some way, and if it does not adopt mainstream linguis-

tics, then it develops its own. That is true of computational linguistics, though it is obscured by the fact that computational linguists focus so heavily on language technology, and ignore fundamental questions about the nature of language. But technology is not the real issue. The greater differences between the fields stem from the philosophical shift in computational linguistics that took place around 1990, already briefly mentioned, that has been called a "statistical revolution." The expression is a little breathless, but it does aptly characterize the profundity and rapidity of the transformation. A similar change swept through the larger field of artificial intelligence at about the same time, with many of the same consequences, though not necessarily the same proximal causes.

When probabilistic models first began to spread through computational linguistics, there was resistance to what were perceived as shallow, brute-force methods, that perhaps offered greater robustness, but considerably less finesse, than familiar logic-based approaches. Understanding the new methods demanded an investment in mathematical retraining: probability and optimization were not areas in which most computational linguists had advanced skills. But, as they say, the reason one visits Manhattan is not because it is easy to get there. Statistical methods were adopted, not because they were easy to master, but because people realized that they were worth the effort. The new methods provided solutions to problems that had frustrated the field for years.

In the late 1980s, two problems were widely identified as critical impediments to progress: ambiguity resolution and portability. Natural language is rife with ambiguities, and ambiguity resolution at the time relied on hand-crafted disambiguation rules. Such rules were not only relatively ineffective—it seemed each new rule created as many problems as it fixed—but they exacerbated the second problem, the problem of portability. The rubric *portability* refers to the difficulty of porting a system developed for one subject domain to a new domain, but the larger issue was how to move beyond limited domains—less charitably, "toy problems"—to unrestricted language. The lack of progress on these two issues had created a palpable frustration.

The disambiguation problem was often illustrated by examples such as the following:

> when he began flailing about, he made her duck
> when he invited her to dinner, he made her duck

Examples like these seemed to show that even modest NLP tasks—here, the determination of the part of speech of "duck"—can require arbitrarily complex reasoning. Because such reasoning was feasible only

in heavily restricted domains, the idea of doing accurate part of speech disambiguation (much less, parsing and understanding) of unrestricted text seemed like science fiction. Such examples also seemed to indicate the hopelessness of breaking the monolithic natural language understanding problem into smaller pieces, like part of speech disambiguation, that could be tackled separately.

But in 1988, two papers appeared that showed how to adapt Hidden Markov Models, probabilistic models that were widely used in speech recognition, to part of speech disambiguation. The papers, Church (1988) and DeRose (1988), treated part of speech disambiguation as a stand-alone problem, separate from parsing and reasoning, and they dealt with unrestricted text. The received wisdom held the task to be impossible, and yet Church and de Rose both reported accuracies of 95–99%. This came as a shock to the community. It seemed to solve the field's two great frustrations, at the cost of performance that was only slightly less than perfect. Probabilistic methods were soon being applied to nearly every problem of natural language processing, and within a few years had reshaped the field.

Awareness of another benefit came more gradually. After the new methods were well established, it became clear that they had also brought a greater scientific maturity to the field. Before 1990 the typical paper in computational linguistics described a piece of software, showing its capabilities with a few illustrative examples. After 1990, the typical paper described an **experiment,** with careful evaluation in the form of quantification of predictive accuracy and comparison to previous experiments addressing the same question. Russell and Norvig put it this way, addressing the same change in AI at large:

> Recent years have seen a revolution in both the content and the methodology of work in artificial intelligence. It is now more common to build on existing theories than to propose brand new ones, to base claims on rigorous theorems or hard experimental evidence rather than on intuition, and to show relevance to real-world applications rather than toy examples . . ..
>
> In terms of methodology, AI has finally come firmly under the scientific method. To be accepted, hypotheses must be subjected to rigorous empirical experiments . . .. Through the use of the Internet and shared repositories of test data and code, it is now possible to replicate experiments. [Russell and Norvig (2002):25–26]

I expect that linguistics will undergo a similar change, but I do not expect it to happen very fast. The hurdles to understanding are much higher than they were in computational linguistics. To be sure, linguistics has shown an increasing interest in computational linguistics; but

I suspect that most linguists see little relevance to their own work. I think they are mistaken, and in the second half of this essay I would like to describe the linguistics that computer scientists have developed, and show not only that it is genuine linguistics, but that it employs superior methods to those used in mainstream linguistics.

First, though, let us look a little further at the history. We will see that the new linguistics is not so entirely new, but to some degree a resumption of older lines of inquiry.

1.1.3 Revolution or counter-revolution?

One must be careful with talk of a "revolution" and a "new linguistics." I recently came across an advertisement for the launch of a journal titled *Statistical Methods in Linguistics (SMIL).* The headline promises "A New Approach to Linguistics," and goes so far as to list communications research, information theory, and stochastical models for syntax as categories of linguistic research. This would seem to be just the kind of new approach that I have in mind—except that the advertisement appears in the 1963 volume of *Economics of Planning.*

I do not doubt that the journal made valuable contributions over its lifetime. (It went out of publication in 1980.) But it did not have a profound effect on the field of linguistics. Is there any reason to believe that the situation is different now?

To modern sensibilities it seems absurd to include communications research and information theory under linguistic research, as the SMIL advertisement did, but the 1950s and early 1960s were a period in which linguistics enjoyed an unusual prominence in mathematics and computation, largely because of the linguistic interests of the founders of the fields of information theory and cybernetics.

Claude Shannon created information theory at Bell Laboratories, in service of coding for the transmission of speech. The seminal paper was "A mathematical theory of communication" (1948). Accompanied by a lengthy general exposition of information theory written by Warren Weaver, it appeared in book form as Shannon and Weaver (1949). A review by Charles Hockett (1953) appeared in *Language,* bringing it to the attention of the linguistics community. The review consists of an accessible summary of the work, and an explicit discussion of its significance for linguistics.

Weaver and Shannon were themselves well aware of the linguistic significance of their work. Shannon followed up the first paper with an explicit application of the theory to human language (1951). Even before that, Weaver wrote a memorandum (1949) proposing to treat machine translation as a cryptographic problem, closely related to the

problems for which Shannon had created information theory. The seed of the memorandum was a letter that Weaver wrote to Norbert Wiener, containing the famous quote:

> When I look at an article in Russian, I say: "This is really written in English, but it has been coded in some strange symbols. I will now proceed to decode." [Weaver (1949): p. 18]

Weaver's correspondent, Norbert Wiener, had just published his influential book *Cybernetics; or, Control and Communication in the Animal and the Machine* (1948). Wiener's central interest was self-regulation, but viewing the problem of self-regulation as one of designing effective feedback essentially reduces regulation to communication among the components of the system. Wiener was very interested in the linguistic applications of the theory: in his view, human language is the means of communication of the components of a social system.

The journal *Information and Control* was founded in service of these two fields—information theory and cybernetics (or control theory)—and linguistics featured prominently in its early issues. For example, the first volume (1957) included not only articles by Claude Shannon on coding theory and communication channels, but also by George Miller and Elizabeth Friedman on applications of Shannon's 1951 paper to English texts; by Miller, Newman, and Friedman on length-frequency statistics for English; and by George Miller and Noam Chomsky on finite-state languages.

To further illustrate the interest of the mathematical community in linguistics, we might also cite a symposium on the *Structure of Language and its Mathematical Aspects,* sponsored and published by the American Mathematical Society [Jakobson (1961)]. It included such titles as "A measure of subjective information" (Rulon Wells) and "Linguistics and communication theory" (Roman Jakobson), not to mention several papers on parsing, such as "Grammar for the hearer" (Charles Hockett).

In short, the "statistical revolution" in computational linguistics was in many ways a counter-revolution, a resurrection of linguistic applications of information theory and control theory that had been abandoned during the "generative revolution" (or "symbolic revolution") of the early 1960s. It is striking, for example, that modern research in machine translation is dominated by the noisy channel model, which takes Weaver's original idea of translation as decoding quite literally.

Also in AI more broadly, the introduction of statistical methods was a resumption of the work from the 1950s. As Russell and Norvig (2002) put it, "AI was founded in part as a rebellion against the limitations of

existing fields like control theory and statistics, but now it is embracing those fields." They go on to quote David McAllester:

> In the early period of AI it seemed plausible that new forms of symbolic computation ... made much of classical theory obsolete. This led to a form of isolationism in which AI became largely separated from the rest of computer science. This isolationism is currently being abandoned. [Russell and Norvig (2002):25]

McAllester explicitly names information theory, control theory, and stochastic modelling as examples of "the rest of computer science."

So why did linguistics turn away from probabilistic models in the first place, and why might things be different now?

One issue, emphasized by Chomsky (1956), is the finite-state nature of the models espoused by Shannon, Weaver, and Wiener. They treated sentences as word strings, and offered no explicit representation of sentence structure. The need for structure is a compelling argument, but not actually very relevant. Probabilistic context-free models were known already in the 1960s, and more recent work has developed probabilistic attribute-value grammars, with even greater generative capacity [Abney (1997)].

A second issue is that probabilistic models are associated in the minds of many with behaviorism, and hence tarred with the same brush. The main criticism against behaviorism is that it refuses, on *a priori* grounds, to consider any internal state which might explain regularities in observables. Hence in particular it can offer no plausible account of language learning, which patently involves the acquisition of rich internal state [Chomsky (1959)]. The focus in early information theory on finite-state models does invite a similar criticism, but the criticism is absurd if applied to probabilistic models more generally. Probabilistic models do not reject internal state, quite the opposite. The fundamental question that probabilistic models address is how to infer internal state correctly from external observables.

Indeed, the need for better learning methods was a central reason that computational linguistics turned to probabilistic models. Generative linguists often indulge in rhetoric about learning, but in computational linguistics, practical and mathematically well-understood learning algorithms are all-pervasive; without exception, those algorithms are deeply rooted in probability and statistics.

The test of a learning method or statistical inference method is how well it predicts the future, in the form of a freshly-drawn sample from the same data source. Systematic study of a system as complex as language requires experiments on the large scale. Thus computational

linguistics came to the approach that I call data-intensive experimental linguistics.

Now Chomsky was right to criticize behaviorism for ignoring internal state, but he went too far in the other direction. As we will see, his conception of "competence" essentially makes everything internal, and nothing objectively observable. Any inconvenient observation can be dismissed as "performance." By contrast, a correct experimental approach distinguishes crisply between that which all observers will agree on, regardless of their theoretical bent, and that which is properly theory- or model-internal; the goal is precisely the inference of internal state from external observations.

In fact, let me contradict myself now and suggest that experimental linguistics is neither a revolution nor a counter-revolution. If information theory is the thesis, and generative grammar is the antithesis, then experimental linguistics is the synthesis. We will see that it is actually very much in the spirit of Chomsky's original formulation of generative grammar. The key difference turns out to be not generativity—probabilistic models are typically generative—but rather a difference in the understanding of the scientific method, and, in particular, of the nature of experimentation and prediction, the relationship between theoretical constructs and observables, and simplicity.

1.2 The scientific method

1.2.1 Description and theory

Let us first consider the approach that Chomsky was reacting against, that of the American structuralists. They were primarily concerned with *description* of languages, which led to considerable attention to the definition of terms, precise descriptions demanding precise terms. Unfortunately, linguistic entities are not so well-behaved as to admit of easy definition. Choose any reasonably concise definition of *noun,* or *phoneme,* or any linguistic concept you will, and one quickly encounters examples where it is unclear how to apply the definition. A great deal of effort went into refining definitions, or replacing them with explicit methods of classification ("discovery procedures"). These survived into generative linguistics in the form of diagnostics, and they arise in contemporary computational linguistics in the form of "style books" or annotation manuals.

The attempt to craft the perfect definition was eventually abandoned, most explicitly in the writings of Zellig Harris. He preserved the key desideratum of **objectivity**—two investigators applying the same methods to the same data should obtain precisely the same results—

and to that end he defined ever more rigorous analytic procedures. But he abandoned the idea that there was one "true" set of methods leading to one "true" set of results. It was naive to talk of "the" set of nouns or phonemes, as if they were Platonic ideals. Two different phonemicizations "differ not in validity but in their usefulness for one purpose or another (e.g. for teaching the language, for describing its structure, for comparing it with genetically related languages.)" [Harris (1951):9, fn. 8]

Chomsky went to the other extreme. For Chomsky, linguistic concepts like noun and phoneme were not defined terms but theory-internal concepts that represented a hidden psychological truth of the matter, connected to data only via the deductive apparatus of the theory as a whole. He was in a sense raising the sights of linguistics: the aim was not merely *description* of languages, but the construction of *scientific theories* of languages.

Moreover, he aimed to go beyond theories of individual languages to theories of language as a whole. Universal linguistics was an old desire, but previous attempts at universal linguistics had devolved into fanciful speculation with little basis in facts, and the structuralists had learned to be cautious. Bloomfield wrote that "the only useful generalizations about language are inductive generalizations," and that universal linguistics, "when it comes, will be not speculative but inductive" [Bloomfield (1933):20]. Chomsky threw caution to the wind, and trusted to the elegance of the theory, leavened by comparing "interesting" predictions of the theory to speakers' judgments, to deliver Truth.

1.2.2 Prediction and testing

The scientific method, as everyone knows from school, is a cycle of formulating a hypothesis, comparing its predictions to data, then formulating a new hypothesis that fits the data better, *ad infinitum.* There is a very important aspect of the scientific method that Chomsky got right. Hypotheses are generated by human insight, not by mechanical procedures. Coming up with genuinely insightful ideas is very hard; defining a general procedure for coming up with insightful ideas is far beyond our grasp. In a real sense, the structuralists were trying to solve a much harder problem than they needed to: by seeking mechanical procedures for defining what were properly theoretical concepts, they were striving to automate theory formation.

But neither the structuralists nor the generativists have understood the second half of the scientific method, namely, making and testing predictions. In generative linguistics, one does speak of the "predictions of the theory." But the predictions are almost never seriously tested,

and untested predictions are nothing more than speculations.

In generative linguistics, when one speaks of testing the predictions of the theory, one usually has in mind something like the following. One has a collection of "interesting" examples with judgments, typically grammaticality judgments. For each example, one can determine whether one's current theory classifies it as grammatical or ungrammatical, and those determinations constitute the predictions of the theory. The predictions are deemed correct if they accord with one's own judgments. Even if they do not, they are not necessarily deemed *incorrect;* rather, one may argue that "performance effects" or other "extraneous factors" account for the discrepancy, or one may decide that one's original judgments were incorrect; perhaps a judgment of "not very good" should be revised to "not really bad," or perhaps one can tweak the word choices or the context to make the example sound better or worse, depending on which way the theory "wants" it to go.

Such a cavalier approach to data amounts to throwing out the baby with the bathwater. The structuralists were wrong to think that central linguistic concepts like noun and phoneme were observational terms, but they were right to be meticulous in their observations and to worry about well-definition. Application of the scientific method requires both theoretical concepts (of which hypotheses are made), and observational terms (of which data is made). "Adjusting" the data to accord with theoretical predictions is dishonest; even if there is no intentional deception, it represents self-deception.

Beyond carelessness with the data, there is another, quite different way in which the usual linguistic practice fails to *test* the predictions of the theory. One can only test a prediction if one does not know what the outcome will be in advance. Devising a theory that gets known facts right does not constitute testing the predictions of that theory.

To make this clearer, let me tell a little story. A researcher named Smith is interested in balls rolling down an inclined plane. He does due diligence as a scholar, and finds a few data points reported in the literature, as follows:

t	1	1	2	4
d	0.5	1	2	7

(1.1)

Here t is time and d is distance traveled. There are conflicting reports about the distance traveled at $t = 1$. Smith decides that the relationship is:

$$d = 2^{t-1} \tag{1.2}$$

He writes a paper, in which he dismisses the first data point ($d = 0.5$ at $t = 1$), and claims that the value at $t = 4$ should actually be

adjusted upward from 7 to 8, attributing the apparent discrepancy to a "friction effect." Thus his theory "perfectly predicts" the data. Another researcher, Jones, responds with a counter-theory:

$$d = \frac{1}{2}t^2 \tag{1.3}$$

He naturally assumes that $d = 0.5$ is the correct value at $t = 1$. He notes that the theories make different predictions at $t = 3$, and determines that $d = 4.5$ at that point, in agreement with (1.3). Smith dismisses the difference between $d = 4.5$ and the prediction $d = 4$ of his own theory (1.2) as measurement error, and asserts that (1.2) is "obviously preferable on grounds of simplicity," since it does not require the "arbitrary constant 1/2."[1]

The mistakes made by the participants in this little drama are plain. Each picks the facts from the literature that fit his theory, and makes adjustments for extraneous effects (linguists call them "performance errors") when it is convenient and seems plausible. At best, isolated data points are examined to test differences in predictions. When arguments based on data falter, appeal is made to a rather subjective notion of simplicity.

Fit to known data points is an important guide in developing a theory, but it does not qualify as a test of the *predictions* of the theory: there is no art in predicting things that have already happened! A proper experiment is one in which the predictions of the theory are determined *before* conducting the experiment; in which predicted values are compared to observed values systematically, over an entire range of points; and in which measurements are made as accurately and objectively as possible.

With a large number of data points, one could discriminate easily between Smith's and Jones's theories. Inconsistencies arising from random measurement errors are no hinderance—that is what statistical hypothesis testing is all about. One does not adjust the data; rather one adjusts the theory by introducing random variables to represent measurement error. In that way one can precisely quantify the probability that the proposed model generates the observed data, assuming that the model accurately characterizes the hidden process that produces the data. Data is always noisy, and repeated measurements will give inconsistent results. But averaging over many measurements allows one to estimate the true values with arbitrary accuracy.

[1] Actually, Jones has the stronger argument where simplicity is concerned, inasmuch as a quadratic function represents a smaller departure from linearity than an exponential function does.

As for systematic error—like friction, in our example—the proper approach is not to fudge the data or dismiss inconsistencies, but to look at the **residuals:** the differences between prediction and observation across the range of data. If there is a systematic effect, it should be reflected in structure in the residuals; say, an increasing shortfall in the observed value of d for increasing values of t. One can then either hypothesize a model for the previously-neglected effect (friction), or one can redesign the experiment to try to minimize the effect; say, by using a smoother surface. But it is wrong to ignore the discrepancy or to appeal to other factors in a haphazard way, when it is convenient to one's argument.

In the terms used in computational linguistics (adopted from machine learning), the experimental data is **test data**, in contrast to the data points used in formulating the theory, which constitute **training data**. The training data is used in formulating the theory, and in choosing values for any open parameters it might have. In the process, one does use **fit** to the training data as a criterion, but one must also be cognizant of the bias that comes with measuring predictive performance against the same data that is used to choose or refine the theory. Doing well on the training data is no guarantee of doing well if one "freezes" the theory, draws a fresh, unbiased test set, and measures performance on that test set. Performance on the training set routinely overestimates performance on the test set; the phenomenon is known as **overfitting**.

The definitive measure of the quality of a theory is how well it predicts what will happen on the entire, infinite **population** of events of interest, for example, events of rolling balls down inclined planes. In machine learning, the error rate on the population as a whole is known as **generalization error**. Error on the test set is used to estimate generalization error. If the test set is a sufficiently large random sample, drawn without bias from the population, then test error provides an unbiased estimate of generalization error. Training error does not. The training sample is biased: it contains data points that the theory was *designed* to do well on.

1.2.3 Simplicity

We have not yet considered the place of simplicity. Typically there is a trade-off between fit and simplicity. For example, there are simpler theories than those of Smith and Jones, but they have such obviously poor fit that they were not even considered. Here is an example:

$$d = t \tag{1.4}$$

A linear function is simpler than either a quadratic (1.3) or exponential (1.2), but, in this example, its fit to the data is plainly worse.[2]

I stated above that it is no great feat to formulate a theory that fits the data. In fact, for any consistent finite data set, one can construct a polynomial whose fit to the data is perfect. Any two arbitrary points can be fit perfectly by a line ($a+bt$), any three points can be fit perfectly by a quadratic equation ($a+bt+ct^2$), any four points can be fit perfectly by a cubic equation ($a+bt+ct^2+dt^3$), and so on. The cost of perfect fit is a complexity that increases with the size of the data set, measuring complexity here by the degree of the polynomial (the largest power of t). For a given data set, we can choose a more complex theory and a better fit, or a poorer fit but a simpler theory.

How do we make the right trade-off between simplicity and fit? Appeal is often made to simplicity in generative linguistics, in many forms: Occam's razor, theoretical elegance, minimality, and the like. But the appeal is almost always made in a subjective, rhetorical way. Moreover, appeal is made to simplicity as a way of *evaluating* a theory. This reflects a basic misunderstanding about the role of simplicity.

Simplicity is not an end in itself. Occam's razor is not an aesthetic principle or a path to Platonic Truth, but an eminently practical rule of thumb. If two theories have comparable fit to the *training* data, but one is clearly simpler than the other, then the simpler one is much more likely to make better predictions on new data—that is, on the *test* data. Simplification is good when it cuts the fat out of a theory, when it eliminates complexities that reflect quirks of the training set rather than real patterns. The measure of a true generalization is how well it predicts test data. As we eliminate complexities in the theory that represent spurious patterns, test error falls. But if we start eliminating complexities that represent genuine patterns, simplification damages predictive power, and test error starts rising again.

The critical point is this: the trade-off between simplicity and fit is used in *formulating* a theory, not in *evaluating* it. Evaluation is strictly in terms of predictive power over the entire population, and is estimated by test error. When we are formulating the theory, test error is unknown, so we use simplicity and fit to the training data as proxies. Both are heuristic indicators of good generalization and low test error, but they are usually in opposition: even "good" simplification usually causes an increase in *training* error. When we formulate the theory, we do the best we can at trading off fit against simplicity. The measure of

[2]Fit can be quantified, for example, as the average squared error of the predictions. For the data set (1.1), equations (1.2) and (1.3) each have a mean squared error of 0.3125, whereas (1.4) has mean squared error 2.3125.

our success is predictive power, quantified by test error.

This is not to say that any single experiment provides a definitive evaluation of a theory. Rather, experiments, done properly, are powerful tools in evaluating and refining a theory. Of course, if a theory is modified in response to an experiment, what was test data for the old theory is training data for the new one, and new experiments are needed to evaluate the new theory: thus, the cycle of refinement and testing that characterizes the scientific method.

1.3 Experimental syntax

How does all of this apply to linguistics? For example, how does this help us develop or evaluate a theory of syntax? If evaluating a theory consists in testing its predictions, we need to identify what kinds of predictions a syntactic theory makes. In generative linguistics, a syntactic theory aims to provide a grammar for each human language, where each grammar assigns a syntactic structure to each sentence of its language. The simplest, most direct experiment, then, is something like the following.

> **Experiment 1.** Let us consider a particular language L, and let G be a grammar of L, representing our theory of L. Draw sentences of the language at random, have people use their judgments as native speakers to determine the syntactic structure of each, and evaluate the grammar by comparing its predictions to the syntax trees that reflect native-speaker judgments.

Of course, that simple statement glosses over a number of very complex issues, but none of them are insuperable. Indeed, the experiment just described is *precisely* the standard approach to evaluating parsers, in computational linguistics. A large sample of sentences, each manually annotated with its syntactic structure, is called a **treebank**. Reasonably large treebanks currently exist for several languages, including English, Czech, Chinese, and Arabic.

In a proper evaluation, a training sample of trees is used to develop the grammar (or parser), and when the grammar is complete, a test sample is drawn to evaluate how well it predicts which tree is assigned (by human judges) to each sentence. In practice, in evaluating "treebank parsers," the treebank is divided into a training set and a test set. The test set is kept strictly hidden during parser development, and used only to evaluate the completed parser. This is legitimate if the training and testing portions are independent samples, and the test set is not examined in any way during parser development.

It may seem odd that the same experiment should be used to test both grammars and parsers. Is this really a direct evaluation of a grammar, or only an indirect evaluation, via a parser based on the grammar?

The critical point is the last phrase of the definition of Experiment 1: "evaluate the grammar by comparing its predictions to the human syntax trees." In order to compare the predictions of the grammar to human judgments, we must determine what those predictions are. Computing a grammar's predictions regarding the syntactic structure of a given sentence is precisely what a parser is for. It is important to understand that, if our interest is in testing the predictions of the grammar, the use of a parsing *program* is merely a convenience—one could in principle compute the predictions of the grammar by hand.

There is precedent for the use of parsing to evaluate grammars. To quote from Chomsky,

> . . . a theory of linguistic structure that aims for explanatory adequacy must contain
>
> (i) a universal phonetic theory that defines the notion "possible sentence"
> (ii) a definition of "structural description"
> (iii) a definition of "generative grammar"
> (iv) a method for determining the structural description of a sentence, given a grammar
> (v) a way of evaluating alternative proposed grammars
>
> [Chomsky (1965):31]

Item (iv) is a parser: Chomsky considered it part and parcel of any adequate linguistic theory. His motivation is evident when we consider his definition of adequacy of a grammar:

> A grammar can be regarded as a theory of a language; it is *descriptively adequate* to the extent that it correctly describes the intrinsic competence of the idealized native speaker. The structural descriptions assigned to sentences by the grammar, the distinctions that it makes between well-formed and deviant, and so on, must, for descriptive adequacy, correspond to the linguistic intuition of the native speaker (whether or not he may be immediately aware of this) in a substantial and significant class of crucial cases. [Chomsky (1965):24]

Apart from some questions about exactly what constitutes "judgments" and "intuitions," and what is meant by "crucial cases," this is essentially our Experiment 1. In particular, to determine whether a grammar is adequate or not, it is necessary to determine what structural description, if any, it assigns to given sentences—hence item (iv).

Following Collins (1999), it is useful to think of a parser as the combination of a grammar with a search algorithm. The grammar defines

what the correct syntactic structure is, but does not give any indication how to construct it. The search algorithm looks for the structure that the grammar deems correct, but sometimes may fail to find it, or finds the wrong structure. For sentence s, let $F(s)$ be the structure *found* by the search algorithm, let $G(s)$ be the structure according to the *grammar,* and let $H(s)$ be the true structure, according to *human* judgment. There is an error if $F(s) \neq H(s)$. But there are two ways an error can arise: a **grammar error** occurs if $G(s) \neq H(s)$, and a **search error** occurs if $F(s) \neq G(s)$.[3] When one uses a treebank to evaluate a parser, one is interested in all errors. When evaluating a grammar, one is interested only in grammar errors.

If our interest is grammatical, we would like to limit or at least quantify the errors due to search. One might eliminate search failures by using exhaustive instead of approximate search. Alternatively, one could at least quantify the two kinds of error by examining cases where $F(s) \neq H(s)$. If $F(s)$ is actually better than $H(s)$ according to the grammar, we have confirmed that the error is indeed a grammar error. If not, we keep running the search algorithm until it either finds $H(s)$ (in which case the original error was a search error) or a structure that the grammar considers better than $H(s)$ (in which case we have a grammar error); though in some cases we may have to give up without an answer after all patience is exhausted. I suspect that most errors are currently grammar errors and not search errors, but as far as I know, the analysis has never been done. It is not a pressing question for parser evaluation, but quite important for grammar evaluation.

Now it may seem mysterious why there should be search at all. The basic issue is this: grammars typically admit many structures—often thousands of structures—for sentences that are unambiguous according to human judgment. This is just as true for the grammars that syntacticians formulate as it is for the grammars that computational linguists formulate. Syntacticians simply never ask the question of how many structures their grammar admits for any given sentence.

The usual method for dealing with this issue is to "soften" the grammar, that is, to use the grammar to define, not a binary grammatical-ungrammatical distinction, but degrees of goodness. A common approach is to factor the grammar into a **well-formedness definition** (for example, a set of rewrite rules), which determines a class of well-formed syntactic structures, and a **weight function** (for example, weights attached to the rewrite rules), which assigns a "degree of good-

[3]In principle, a search error could cancel a grammar error; we neglect that case as vanishingly rare.

ness" $g(\sigma, s)$ to any given structure σ for sentence s.

In this approach, we define $G(s)$ to be the structure σ that maximizes $g(\sigma, s)$. The grammar makes it easy to compute $g(\sigma, s)$ for a given structure σ, but it does not indicate how to find the best structure. The search algorithm searches through structures, computing $g(\sigma, s)$ for each, and returns the best one it finds. The number of possible structures increases rapidly with sentence length, so that an exhaustive search is rarely possible, hence the search algorithm may fail to find the structure that actually maximizes $g(\sigma, s)$.

The idea behind the weight function is quite similar to the idea that motivated Chomsky's item (v) in the earlier citation. A linguistic theory may well provide multiple grammars that are consistent with a given set of primary data, and Chomsky's evaluation function selects one grammar out of that set as the grammar assigned to the language by the theory. Similarly, for a given sentence, even an unambiguous sentence, there may well be multiple structures admitted by the well-formedness definition. The weight function selects one to be *the* structure of the sentence according to the grammar.

All of this is predicated on the assumption that typical grammars generate a great deal of spurious ambiguity, so a closer examination of that assumption is in order. Consider a sentence like *my dog barks*. This sentence has one obvious interpretation. If one were to ask a random speaker of English whether it is ambiguous or not, the answer would surely be "no." But when well-formedness is defined in the natural way, there is also a second well-formed structure, in which *my dog barks* is a noun phrase. A "bark" is a kind of ship, hence a "dog bark" is predicted to be a bark that is associated in some way with dogs. (For further discussion and more examples, see Abney (1995).)

When confronted with this property of his or her well-formedness definition, a syntactician's natural response may be to revise the facts, and decide that *my dog barks* is ambiguous after all. All the speakers who judged it to be unambiguous were mistaken in their judgment. That kind of response goes back to Chomsky, and in fact to the rather cryptic parenthetical "whether or not [the speaker] may be immediately aware of [his own intuitions]" in the previously cited passage. Chomsky argued that an investigator may need to apply considerable ingenuity to determine the facts about a particular sentence: "it may be necessary to guide and draw out the speaker's intuition in perhaps fairly subtle ways before we can determine what is the actual character of his knowledge of his language" [Chomsky (1965):24]. In the case of *my dog barks,* reminding the speaker of the more obscure meaning of *bark* may "make him aware" that he "actually knows" the sentence to be ambiguous,

even though his first impression was that it was unambiguous.

This response to the issue is fundamentally misguided. The urge to revise the judgments is exactly the kind of cavalier attitude toward data that we ought at all costs to avoid. No native speaker of English would consider *my dog barks* to be ambiguous. If there is a tendency to revise judgments after being prompted with other possibilities, that is an observation worth accounting for, but it neither explains nor discounts the original observation. If the phrase is genuinely ambiguous, but speakers are not very good at realizing when something is ambiguous, we would expect some percent of the speakers to judge that it is unambiguously a sentence, and the remainder to judge that it is unambiguously a noun phrase. But that is not the case.

An explanation that makes sense of both observations is that judgments of sentence structure are sensitive to context. We might consider refining $H(s)$ to be sensitive to relevant aspects of the context, and replace it with $H(s, c)$, the assignment of a structure to a given utterance s in a given context c. In the "neutral" context, the utterance *my dog barks* is interpreted as a sentence, but in a context in which we are discussing archaic sailing vessels and canine cargo, it is judged to be a noun phrase. Critically, it is *unambiguous* in both contexts, though it unambiguously has one structure in one context and unambiguously has a different structure in a different context. It is only when the interrogator brings both contexts to mind, but gives no instructions as to which context the informant is to imagine him- or herself placed in, that a judgment of ambiguity arises.

If this line of reasoning is correct, genuine ambiguity is much rarer than we have come to believe. Indeed, it is universal practice in treebanks to assume that there is no genuine ambiguity in context; a unique structure is assigned to each sentence. There are many uncertainties that arise when annotating sentences for a treebank, but ambiguity does not appear to be a significant problem. Human annotators seem to have little trouble saying which of two potential readings is correct for a given sentence, particularly since the sentences in treebanks are not isolated, but occur in context.

Even the dependency on context may be less than commonly assumed. Current parsers ignore context. To the extent that they do reasonably well, they provide an indication that contexts that override the "default" judgment are rare in practice.

Puns and other forms of wordplay are cases in which a sentence is intentionally ambiguous. But wordplay is just that: *play*. If there were rampant ambiguity, hence rampant indeterminacy about the structure that a speaker intended, language could hardly be used for communi-

cation.

The point bears amplification. Anyone who has written a grammar for an automatic parser can attest that even garden-variety sentences end up with hundreds or thousands of syntactic structures, according to the grammar. To be very conservative, suppose there are only ten parses on average. The linguistic theory represented by that grammar predicts that, even when the hearer perfectly understands the words that were spoken, he or she arrives at the wrong interpretation *nine times out of ten.* Such a linguistic theory—and every current linguistic theory is such a theory—fails dramatically to explain how communication is even possible.

The impression of rampant ambiguity is a product of poor observational methods. An experiment in which speakers make judgments in a default context, or an experiment involving unusual contexts, are both valid experiments. But one must fix the experimental design in advance, and make observations objectively. The result of an observation cannot depend on the ingenuity of the interrogator. The ingenuity of the researcher is of tremendous importance in theory formation, and in designing experiments, but has no place at the point when an observation is made that provides a data point in an experiment.

Another objection that Chomsky raises against experiments like Experiment 1 is that speaker judgments are not the object of linguistic study; the object of study is the knowledge that underlies those judgments. That is true, but it does not follow that the judgments are therefore "less important" than internal state. Our goal is indeed to construct a model of what is going on internally to produce the observables. But the only way we can determine if our guesses about the internals are correct is via their predictions about observables. Making meticulous, systematic observations is critically important for any valid inference about internal state. Using one's model to replace observations with imputations, as in the example of the "ambiguity" of *my dog barks*, leads not to insight but to self-deception.

Another potential objection is that characterizing human judgments, as opposed to "idealized judgments" of the Chomskyan sort, puts one in the position of dealing with non-syntactic things like contexts and meanings, even to answer a simple question of syntactic structure. So be it. One can neglect the effects of context at the cost of a poorer fit to the data, or one can design experiments that minimize the effects of contexts. But in either case, we admit that something like Experiment 1 is the real aim, even if a fully successful model is unachievable with syntactic constraints alone.

An appeal to "competence and performance" might be appropriate if the effects in question were negligible, but in this case the divergence between prediction and observation is massive; hiding behind "performance effects" would be a bald abdication of responsibility. One could be responsible to the facts by offering a combined theory of competence and performance, but a "competence theory" standing alone is untestable speculation.

It is not as if workable models of "performance" are all that hard to construct. The weighted-grammar approach described above is well understood and reasonably effective.

The question posed by Experiment 1, namely, the accuracy of a grammar's predictions regarding the syntactic structures assigned to sentences, seems as basic and elementary a question as one could ask of syntax, even more basic than the question of grammaticality, which, though simpler in the sense of admitting a yes/no answer, has always struck me as derivative. My first exposure to linguistics was the exercise of diagramming sentences in grade school. The least one can expect of an explicit generative grammar is that it assign the correct diagram to run-of-the-mill sentences. Willingness to step up to that challenge is probably the single most important difference between mainstream syntax and the data-intensive experimental approach.

1.4 Experimental universal linguistics

I stated in the previous section that simplicity had a role in formulating grammars but not in evaluating grammars. That is true, but there is something more to say about it. Let us consider X-bar theory, which represents a classical case of a theoretical revision motivated by simplicity. The motivation is *not* that conforming to X-bar theory improves the predictive power of the grammar of English, or of any one particular language. Rather, X-bar theory is intended to improve the predictive power of universal grammar. To continue the previously cited passage from Chomsky:

> To the extent that a linguistic theory succeeds in selecting a descriptively adequate grammar [for each language] on the basis of primary linguistic data, we can say that it meets the condition of *explanatory adequacy*. [Chomsky (1965):25]

In short, linguistic theory defines the function computed by the human language learning algorithm. For each sample S drawn from a language (taking a sample to represent Chomsky's *primary linguistic data*), the theory defines a unique grammar $G(S)$—but it does not necessarily give concrete instructions for how to go about constructing $G(S)$ given

S as input.

Simplicity has a role in universal linguistics, above and beyond its role in grammar formulation. At the same time, the role of simplicity in universal linguistics is exactly analogous to its role in a single language. It is not an end in itself, but rather a means to reduce generalization error. By observing commonalities across grammars, we hope to capture generalizations that hold in the population of grammars. At the same time, we hope to avoid spurious generalizations, in which we mistake accidental similarities of the grammars we have studied for generalizations that hold throughout the population. Simplifying grammars consists in factoring out commonalities. It is good if it results in improved prediction on a freshly-drawn test set (of languages to be learned), and bad if not.

The kind of experiment we are implicitly assuming is something like the following.

> **Experiment 2.** Choose a sample of languages at random. For each language, determine what grammar one's universal linguistic theory predicts. Conduct Experiment 1 on that grammar to evaluate it. A predicted grammar is correct if it makes correct predictions regarding the structure of sentences of the language, and a theory is correct if it predicts grammars correctly, given language samples.

This experiment is admittedly much harder to carry out. Effectively, it represents the ultimate goal of experimental linguistics.

Fully carrying out Experiment 2 requires one to study language learning in a more concrete way than is commonly done in linguistics. As discussed earlier, computational linguists have made learning a central feature of their approach to language. But unlike parsing, for which imperfect but useful solutions exist, work in language learning has focused on learning smaller slices of lexical and grammatical information. Usable end-to-end language learning systems do not exist. Unlike parsing, unsupervised learning of complete grammars—the technical term is **grammatical inference**—is not of high priority in the pursuit of technology, but it is of profound importance for linguistics. It represents an area of opportunity for data-intensive experimental linguistics, and indeed some of the most compelling work in the area is being pursued by linguists; the work of John Goldsmith comes particularly to mind [e.g., Goldsmith (2001)].

Until effective grammatical inference methods are available, manual determination of the grammar assigned to a language by the theory is an alterative. But even manual determination requires a more explicit definition of the function $G(S)$ than current linguistic theories offer.

At the very least, the simultaneous development of grammars for treebanks in multiple languages should give some indication of which metagrammatical generalizations, of the X-bar theory ilk, make it easier to create grammars that do well in the sense of Experiment 1, and which generalizations are driven by a misguided sense of aesthetics.

Unfortunately, even carrying out Experiment 1 on multiple languages is a tall order. A typical treebank contains on the order of a million words of text, and requires many person-years of effort to create. One option is to explore methods of reducing the effort of treebanking, and to explore methods of maximizing the utility of smaller treebanks. This is another topic of importance to experimental linguistics that has not received much attention in computational linguistics.

In parallel, it is worth asking whether there are experiments that can be used to evaluate syntactic structures without access to treebanks. Not only are treebanks extremely labor-intensive to produce, but syntactic trees are really not "observables." Syntactic structures are very much theory-internal objects. In practical terms, it is impossible to get much agreement concerning syntactic structures between adherents of different schools of syntax.

One possibility is to exploit translations. After all, at root, linguistic structure is about the relationship between sound and meaning. Taking that aphorism literally, the purpose of syntactic structure is to mediate the translation from a sentence (a sequence of sounds or textual representations of sounds) to a meaning. So a good syntactic model is one that assigns the right meanings to sentences. Now, writing down meanings is even less practical than writing down syntactic structures. But if the sentence of interest is a sentence in Russian, say, then the translation into English is quite a serviceable representation of its meaning, for an English speaker. This leads to Experiment 3.

> **Experiment 3.** Choose a sample of languages at random. For each language, determine what grammar one's universal linguistic theory predicts. For each language, collect a sample of sentences, along with their translations into a reference language (for example, English). A predicted grammar is correct if its predictions regarding translations are correct.

There is some existing work along these lines. For example, David Yarowsky and his students have developed methods for **cross-language bootstrapping**, which is the transferring of tools such as part of speech taggers from one language to another [e.g., Yarowsky et al (2001)]. The methods assume a source language for which, say, a part of speech tagger is available, a target language of interest, and a collection of

unnannotated texts in the target language with translations in the source language. Given texts with translations, methods exist for determining word-to-word alignment: that is, which source-language word corresponds to each target-language word. One makes a provisional assumption that words and their translations have the same part of speech more often than not; using word alignments to project parts of speech from source to target language yields an initial, noisy part-of-speech annotated text in the target language. Statistical methods are then applied to improve the quality of the annotated text.

A special case of translation is represented by the glosses in interlinear glossed text. Interlinear glossed text has a long history in documentary and descriptive linguistics, and as a translated text it has the advantage of including (typically) not only a colloquial translation but morpheme-by-morpheme glosses. Thus it provides more detailed alignment information than the automatic word alignments usually used in cross-language bootstrapping. A natural idea is to parse the gloss text, and use the morpheme-level alignments to construct the syntactic structure of the sentence in the original language, on the assumption that the English gloss has the same syntactic structure as the original sentence. For pioneering work along these lines, see Xia and Lewis (2007).

1.4.1 A universal corpus

Whatever the precise form of the experiments, any experimental foray into universal linguistics will be a data-intensive undertaking. It will require substantial samples of many languages—ultimately, all human languages—in a consistent form that supports automated processing across languages. The construction of such a dataset—effectively, a **universal corpus**—is admittedly a very ambitious project, but it is indispensible. It is distressing that no such dataset is currently under construction.[4]

The construction of a universal corpus is particularly urgent given the rate at which languages are becoming extinct. Whether one is interested in experimental universal linguistics or not, a detailed record of every endangered language is clearly an urgent need.

For the study of fully explicit language acquisition algorithms, traditional language descriptions consisting of a grammar and a lexicon,

[4]The Linguistic Data Consortium (ldc.upenn.edu) contains data from dozens of languages, but the selection of languages appears to correlate with their importance for commerce and intelligence, not universal linguistics. The Rosetta Project (rosettaproject.org) is driven by universal linguistics, but it focusses on page images and traditional methods, not automated cross-linguistic processing.

supplemented perhaps with a handful of short texts, is inadequate. A grammar and lexicon represents the *output* of a language acquisition algorithm, not its input. Moreover, traditional grammars and lexica even fall short as characterizations of the output of acquisition. The target of learning is a compact representation of a complete language. Chomsky would call it *knowledge of language* but we may more neutrally call it a **digitization** of a language. The measure of adequacy is whether it supports full comprehension and production of the language (which may be approximated by translation into and out of the language). I consider it unlikely that a human could obtain real competence in a language using only a traditional grammar and lexicon. One cannot obtain fluency without primary data—text and recordings. If so, a traditional grammar and lexicon represents at best a *partial* digitization of a language.

As a practical matter, traditional grammars and lexica completely fail to support computer-assisted research into universal linguistics, or even computer-assisted quality control. With few exceptions, they are available only in print format, pose serious challenges for optical character recognition, and adhere to no standards, not even at the level of character representation, that might support cross-linguistic processing.

In short, if we are ever to move beyond speculation and "toy" models of acquisition to serious models for the acquisition of entire languages, We require a dataset containing, at a minimum, a significant quantity of primary data in the form of speech or text. Grammars, lexica, paradigms, and examples elicited for specific purposes are all useful supplements, but do not supplant the need for primary data. And both for the development and testing of models of acquisition, we require such data for a wide range of typologically diverse languages.

1.5 Conclusion

It is daunting to contemplate constructing a universal corpus and bringing into linguistics the kind of computational sophistication necessary to formulate and evaluate explicit, full-scale models of language acquisition. But it is unavoidable if the field is to move beyond *a priori* theory construction supported at best by a handful of examples, to a systematic, scientific study of universal linguistics.

Linguistics is not alone in the evolution of data-intensive methods. The development of a universal corpus is no more ambitious than the Sloan Digital Sky Survey, the development of climatological models, or the Human Genome Project. Consider this description of the latter:

> One of the greatest impacts of having the sequence may well be in enabling an entirely new approach to biological research. In the past, researchers studied one or a few genes at a time. ... [Now] they can approach questions systematically and on a grand scale. They can study ... how tens of thousands of genes and proteins work together in interconnected networks to orchestrate the chemistry of life. [Human Genome Project (2008)]

Emerging from computational linguistics is a new approach to linguistic research that is predicated on systematicity and experimentation enabled by large-scale data collection. Language is a computational system, and there is a depth of understanding that is unachievable without a thorough knowledge of computation. But even more than that, the new approach reflects a deeper understanding of the scientific method, and places linguistic inquiry firmly within the paradigm of data-intensive research that has come to characterize modern sciences.

Acknowledgements

I would like to thank Mark Johnson, Joe Pater, Robin Queen, two anonymous reviewers, and the participants in the University of Michigan CompLing discussion group (Ezra Keshet, Kevin McGowan, Arzucan Özgür, Evelyn Richter, Terry Szymanski, Richmond Thomason, Stephen Tyndall) for very helpful comments on an earlier draft.

References

Abney, Steven. 1997. Stochastic attribute-value grammars. *Computational Linguistics* 23(4):597-618.

Abney, Steven. 1995. Statistical methods and linguistics. in Klavans J, Resnik P (eds), *The Balancing Act.* The MIT Press. Cambridge, MA.

Automatic Language Processing Advisory Committee (ALPAC). 1966. *Languages and machines: Computers in translation and linguistics.* Publication 1416, National Research Council, National Academy of Sciences.

Bloomfield, Leonard. 1933. *Language.* Holt. New York.

Noam Chomsky. 1956. Three models for the description of language. *IRE Transactions on Information Theory.* IT-2(3):113–124. Institute of Radio Engineers, New York.

Noam Chomsky. 1959. Review of *Verbal behavior* by B.F. Skinner. *Language* 35.1.

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

Church, Kenneth. 1988. A stochastic parts program and noun phrase parser for unrestricted texts. *Proceedings of the Second Conference on Applied Natural Language Processing.* Austin, Texas.

Collins, Michael. 1999. *Head-Driven Statistical Models for Natural Language Parsing.* Doctoral dissertation, Univ. of Pennsylvania.

DeRose, S. 1988. Grammatical category disambiguation by statistical optimization. *Computational Linguistics* 14(1).

Goldsmith, John. 2001. Unsupervised learning of the morphology of a natural language. *Computational Linguistics* 27(2):153–198.

Harris, Zellig. 1951. *Structural Linguistics.* University of Chicago Press. Chicago.

Hockett, Charles. 1953. Review of The Mathematical Theory of Communication by Claude Shannon and Warren Weaver. *Language* 29(1): 69–93.

Human Genome Project. 2008. The science behind the Human Genome Project. www.ornl.gov/sci/techresources/Human_Genome/project/info-.shtml, accessed 2009 Jul 27.

Jakobson, Roman (ed). 1961. *Structure of Language and its Mathematical Aspects.* Proc. of Symposia in Applied Mathematics, vol. XII. American Mathematical Society. Providence, RI.

Kay, Martin. 2005. A Life of Language. *Computational Linguistics,* 31.4.

Russell, Stuart J., and Peter Norvig. 2002. *Artificial Intelligence: A Modern Approach* (2nd Edition). Prentice Hall.

Shannon, Claude E. 1948. A mathematical theory of communication. *Bell System Technical Journal* 27(3-4): 379–423, 623–656.

Shannon, Claude E. 1951. Prediction and entropy of printed English. *Bell System Technical Journal* 30: 50–64.

Shannon, Claude E. and Warren Weaver. 1949 *The Mathematical Theory of Communication.* University of Illinois Press. Urbana and Chicago.

Sparck Jones, Karen. 2002. Some notes on ACL history. `http://www.aclweb.org/archive/misc/History.html`. Created 1994, revised 2002, accessed 2009 Jun 11.

Sproat, Richard. Last updated 2005, accessed 2009 Jul 23. What is computational linguistics? `http://www.aclweb.org/archive/misc/what.html`.

Thesaurus Linguae Graecae. 2009. About the TLG. www.tlg.uci.edu/about, last modified 2009 Mar 18, accessed 2009 Jul 27.

Weaver, Warren. Typescript, July 1949. Translation. Reprinted in Locke, William N., and A. Donald Booth (eds) *Machine translation of languages: fourteen essays.* (Technology Press of the Massachusetts Institute of Technology, Cambridge MA, 1955.)

Wiener, Norbert. 1948. *Cybernetics; or, Control and Communication in the Animal and the Machine.* J. Wiley. New York.

Xia, Fei and William D. Lewis. 2007. Multilingual structural projection across interlinearized text. *Proceedings of the North American Association of Computational Linguistics (NAACL).*

Yarowsky, D., G. Ngai, and R. Wicentowski. 2001. Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of HLT 2001, First International Conference on Human Language Technology Research.*