*L*inguistic *I*ssues in *L*anguage *T*echnology – *LiLT*
Volume 2, Issue 4 May 2007

A Pendulum Swung Too Far

Kenneth Church

Published by CSLI Publications

A Pendulum Swung Too Far

Kenneth Church, *Johns Hopkins University*

1.1 Motivation for Pragmatism

The revival of empiricism in the 1990s was an exciting time. We never imagined that that effort would be as successful as it turned out to be. At the time, all we wanted was a seat at the table. In addition to everything else that was going on at the time, we wanted to make room for a little work of a different kind. We founded SIGDAT[1] to provide a forum for this kind of work. SIGDAT started as a relatively small Workshop on Very Large Corpora in 1993 and later evolved into the larger EMNLP Conferences. At first, the SIGDAT meetings were very different from the main ACL conference[2] in many ways (size, topic, geography), but over the years, the differences have largely disappeared. It is nice to see the field come together as it has, but we may have been too successful. Not only have we succeeded in making room for what we were interested in, but now there is no longer much room for anything else. Figure 1 illustrates the dramatic shift from Rationalism to Empiricism with no end in sight.

According to Hall et al. (2008), the shift started in 1988 with Brown et al. (1988) and Church (1988). Hall et al. (2008) came to this conclusion based on an analysis of the ACL Anthology, a collection of 16,500 published papers in Computational Linguistics from the 1970s to the

[1] http://www.cs.jhu.edu/~yarowsky/sigdat.html
[2] http://www.aclweb.org

LiLT Volume 2, Issue 4, May 2007.
A Pendulum Swung Too Far.

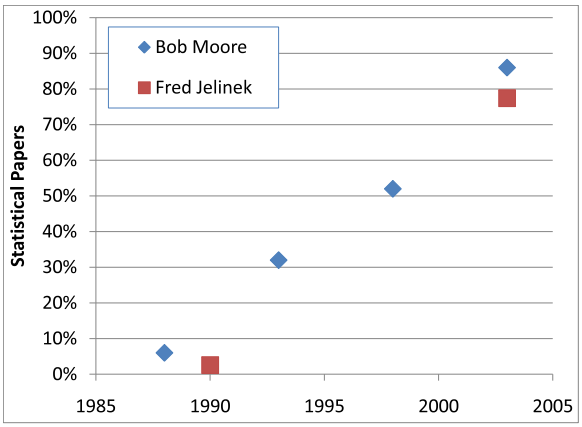

FIGURE 1 The shift from Rationalism to Empiricism is striking (and no longer controversial). This plot is based on two independent surveys of ACL meetings by Bob Moore and Fred Jelinek (personal communication).

present.[3]

However, if we consider a larger time window that goes back well before the ACL Anthology, as illustrated in Figure 2, we see a very different picture. The more salient trend is the oscillation between Rationalism and Empiricism and back with a switch every couple decades:

- 1950s: Empiricism (Shannon, Skinner, Firth, Harris)
- 1970s: Rationalism (Chomsky, Minsky)
- 1990s: Empiricism (IBM Speech Group, AT&T Bell Labs)
- 2010s: A Return to Rationalism?

This paper will review some of the rationalist positions that our generation rebelled against. It is a shame that our generation was so successful that these rationalist positions are being forgotten (just when they are about to be revived if we accept that forecast). Some of the more important rationalists like Pierce are no longer even mentioned in currently popular textbooks. The next generation might not get a chance to hear the rationalist side of the debate. And the rationalists have much to offer, especially if the rationalist position becomes more popular in a few decades.

What motivated the revival of empiricism in the 1990s? What were we rebelling against? The revival was driven by pragmatic considerations. The field had been banging its head on big hard challenges like AI-complete problems and long-distance dependencies. We advocated a

[3] http://aclweb.org/anthology-new/

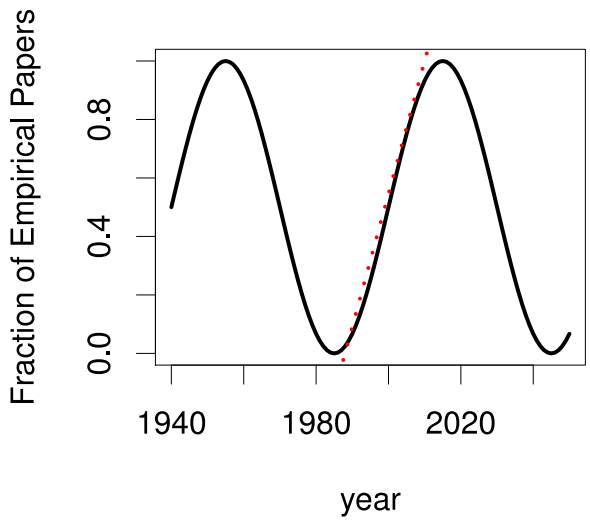

FIGURE 2 An extreme view of the literature, where the trend in Figure 1 (denoted by a dashed red line) is dominated by the larger oscillation every couple of decades. Note that that line is fit to empirical data, unlike the oscillation which is drawn to make a point.

pragmatic pivot toward simpler more solvable tasks like part of speech tagging. Data was becoming available like never before. What can we do with all this data? We argued that it is better to do something simple (than nothing at all). Let's go pick some low hanging fruit. Let's do what we can with short-distance dependencies. That won't solve the whole problem, but let's focus on what we can do as opposed to what we can't do. The glass is half full (as opposed to half empty).

> The 1990s have witnessed a resurgence of interest in 1950s-style empirical and statistical methods of language analysis. Empiricism was at its peak in the 1950s, dominating a broad set of fields ranging from psychology (behaviorism) to electrical engineering (information theory). At that time, it was common practice in linguistics to classify words not only on the basis of their meanings but also on the basis of their co-occurrence with other words. Firth (1957), a leading figure in British linguistics during the 1950s, summarized the approach with the memorable line: "You shall know a word by the company it keeps." Regrettably, interest in empiricism faded in the late 1950s and early 1960s with a number of significant events including Chomsky's criticism of n-grams in Syntactic Structures (Chomsky, 1957) and Minsky and Papert's criticism of neural networks in Minsky and Papert (1969).
>
> Perhaps the most immediate reason for this empirical renaissance is

> the availability of massive quantities of data: more text is available than ever before. Just ten years ago, the one-million word Brown Corpus Francis and Kucera (1982) was considered large, but even then, there were much larger corpora such as the Birmingham Corpus (Sinclair (1987) and Sinclair et al. (1987)). Today, many locations have samples of text running into the hundreds of millions or even billions of words.... The data-intensive approach to language, which is becoming known as Text Analysis, takes a pragmatic approach that is well suited to meet the recent emphasis on numerical evaluations and concrete deliverables. Text Analysis focuses on broad (though possibly superficial) coverage of unrestricted text, rather than deep analysis of (artificially) restricted domains.[4]

1.2 Winters

The research community found the pragmatic approach attractive at that point in time (early 1990s) because the field was in the midst of a severe funding winter, what is now known as the second AI Winter of 1987-93. After yet another cycle of funding busts, the community was relatively receptive to a new approach that promised reliable results that we could bank on. According to Wikipedia[5]

> In the history of artificial intelligence, an AI winter is a period of reduced funding and interest in artificial intelligence research. The process of hype, disappointment and funding cuts are common in many emerging technologies (consider the railway mania or the dot-com bubble), but the problem has been particularly acute for AI. The pattern has occurred many times:
>
> - 1966: the failure of machine translation,
> - 1970: the abandonment of connectionism,
> - 1971-75: DARPA's frustration with the Speech Understanding Research program at Carnegie Mellon University,
> - 1973: the large decrease in AI research in the United Kingdom in response to the Lighthill Report,
> - 1973-74: DARPA's cutbacks to academic AI research in general,
> - 1987: the collapse of the Lisp machine market,
> - 1988: the cancellation of new spending on AI by the Strategic Computing Initiative,
> - 1993: expert systems slowly reaching the bottom,
> - 1990s: the quiet disappearance of the fifth-generation computer project's original goals, and the generally bad reputation AI has had since.
>
> The worst times for AI were 1974-80 and 1987-93. Sometimes one or

[4] Church and Mercer (1993)
[5] http://en.wikipedia.org/wiki/AI_winter

> the other of these periods (or some part of them) is referred to as "the" AI winter.

The busts (winters) often followed a boom, a period of excessive optimism such as:[6]

> Within the very near future—much less than twenty-five years—we shall have the technical capability of substituting machines for any and all human functions in organizations. Within the same period, we shall have acquired an extensive and empirically tested theory of human cognitive processes and their interaction with human emotions, attitudes and values.

We're feeling more confident these days than we felt at the depths of the second AI Winter. 15 years of picking low hanging fruit has produced a relatively stable stream of results, and relatively stable funding, at least when compared to the AI Winters.

1.3 Pierce, Chomsky & Minsky (PCM)

Needless to say, many of the great rationalists that we rebelled against, like Pierce, Chomsky and Minsky (henceforth PCM), would not be happy with where the field is today. Of course, on the other hand, many of the leaders of the field today would not be happy with a revival of their positions. When one of the current leaders of the field heard about this paper, he quipped, "What does Pierce have to offer us today?" PCM's arguments were controversial at the time and remain so because they caused a number of severe funding winters in a number of fields: Speech, Machine Translation and Machine Learning.

This paper is more interested in the common threads among PCM, but it is important to mention that they do not speak with one voice. There is considerable disagreement over Information Theory. Pierce (1961) has nice things to say about both Shannon and Chomsky, even though Chomsky is rebelling against much of Shannon's work on Information Theory. Apparently, these views don't fall neatly into simple equivalence classes (e.g., Rationalists and Empiricists), with perfect agreement within classes, and perfect disagreement across classes.

There is also considerable disagreement over intelligence. Minsky was a founding father of Artificial Intelligence (AI), and Pierce was one of its more outspoken critics: "Artificial intelligence is real stupidity."[7] Pierce objected to anything that attempts to come close to human intelligence, including of course Artificial Intelligence, but also Machine Translation and Speech Recognition. Pierce chaired the (in)famous ALPAC report,

[6] Simon (1960)

[7] http://en.wikipedia.org/wiki/John_R._Pierce

which is largely credited with a funding winter in Machine Translation.[8] Pierce also wrote "Whither Speech Recognition" a controversial letter to JASA (Journal of the Acoustical Society of America) that had a chilling effect on funding for speech recognition.[9]

This paper is more interested in the common threads than the differences. PCM challenged a number of empirical methods that were popular at the time, and have since been revived. Their objections have implications for many popular contemporary methods including Pattern Matching, Machine Learning (Linear Separators), Information Retrieval (Vector Space Model), Language Modeling (ngrams) and Speech Recognition (Hidden Markov Models (HMMs) and Conditional Random Fields (CRFs)).

Students need to learn how to use popular approximations effectively. Most approximations make simplifying assumptions that can be useful in many cases, but not all. For example, ngrams can capture many dependences, but obviously not when the dependency spans over more than n words. Similarly, linear separators can separate positive examples from negative examples in many cases, but not when the examples are not linearly separable. Many of these limitations are obvious (by construction), but even so, the debate, both pro and con, has been heated at times. And sometimes, one side of the debate is written out of the textbooks and forgotten, only to be revived/reinvented by the next generation.

Chomsky wrote about limitations with ngrams and Minsky wrote about limitation with linear separators. Others have written about limitations with other approximations. Tukey (1977), for example, teaches effective use of regression. Tukey encourages students to test for deviations from various normality assumptions. Outliers are a common source of trouble for regression, as are bowed residuals. Many workarounds have been proposed. A common trick is to transform the data with a non-linear transform such as a log. These tricks transform the problem into another problem with fewer troublesome deviations from the assumptions.

1.3.1 Chomsky's Objections

As mentioned above, Chomsky showed that ngrams cannot learn long-distance dependencies. While that might seem obvious in retrospect, there was a lot of excitement at the time over the Shannon-McMillan-Breiman Theorem,[10] which was interpreted to say that, in the limit, un-

[8] Pierce et al. (1966)
[9] Pierce (1969, 1970)
[10] http://cm.bell-labs.com/cm/ms/what/shannonday/shannon1948.pdf

der just a couple of minor caveats and a little bit of not-very-important fine print, ngram statistics are sufficient to capture all the information in a string (such as an English sentence). Chomsky realized that while that may be true in the limit, ngrams are far from the most parsimonious representation of many linguistic facts. In a practical system, we will have to truncate ngrams at some (small) fixed k (such as trigrams or perhaps 5-grams). Truncated ngram systems can capture many agreement facts, but not all.

We ought to teach this debate to the next generation because it is likely that they will have to take Chomsky's objections more seriously than we have. Our generation has been fortunate to have plenty of low hanging fruit to pick (the facts that can be captured with short ngrams), but the next generation will be less fortunate since most of those facts will have been pretty well picked over before they retire, and therefore, it is likely that they will have to address facts that go beyond the simplest ngram approximations.

Center-Embedding

Chomsky not only objected to ngrams, but he also objected to finite-state methods, which include currently popular methods such as Hidden Markov Models (HMMs)[11] and Conditional Random Fields (CRFs).[12]

Finite-state methods go beyond ngrams. Not only can finite-state methods capture everything that ngrams can capture, but they can do more. Unlike ngrams, finite-state grammars can capture some dependences that go beyond n words. For example, the following grammar expresses subject-verb agreement. Nouns and verbs should agree; both should be singular (sg) or both should be plural (pl). With a grammar such as the following, it is possible to capture such a dependency, even if it spans over more then n words.

$$S \rightarrow S_{sg}$$
$$S \rightarrow S_{pl}$$
$$S_{sg} \rightarrow NP_{sg} VP_{sg}$$
$$S_{pl} \rightarrow NP_{pl} VP_{pl}$$
$$NP_{sg} \rightarrow \cdots N_{sg} \cdots$$
$$NP_{pl} \rightarrow \cdots N_{pl} \cdots$$
$$VP_{sg} \rightarrow \cdots V_{sg} \cdots$$
$$VP_{pl} \rightarrow \cdots V_{pl} \cdots$$

[11] http://en.wikipedia.org/wiki/Hidden_Markov_model
[12] http://en.wikipedia.org/wiki/Conditional_random_field

TABLE 1 The Chomsky Hierarchy.

Type	Grammars	Automata
3	Regular	Finite-State Machines (FSMs)
2	Context-Free (CF)	Push-Down Automata (PDA)
1	Context-Sensitive (CS)	Linear Bounded Automata (LBA)
0	Recursively Enumerable	Turing Machines (TMs)

The big question is whether this grammar requires infinite memory. To make this debate rigorous, Chomsky introduced center-embedding and what has since become known as the Chomsky Hierarchy.

The Chomsky Hierarchy has been hugely influential, not only in linguistics, but in many other fields as well, such as Computer Science.[13] Knuth admits to having read Chomsky (1957) during his honeymoon in 1961 and found it to be "a marvelous thing: a mathematical theory of language in which I could use a computer programmer's intuition."[14]

Chomsky showed that there is a simple relationship between the Chomsky Hierarchy and generative capacity:

$$Type\ 0 > Type\ 1 > Type\ 2 > Type\ 3$$

$$Recursively\ Enumerable > CS > CF > Regular$$

In particular, context-free grammars can do more than regular grammars; there are things that can be done with infinite memory (a stack) that cannot be done with finite memory. Chomsky established center-embedding as the key difference between context-free and finite-state. That is, if (and only if) a grammar is center-embedded, then it requires infinite memory (a stack). Otherwise, it can be processed with finite memory (a finite-state machine).

More formally, a grammar is center embedded if there is a non-terminal A that can generate xAy where both x and y are non-empty. If either x or y are empty, then we have the simpler case of left-branching and right-branching. The simpler cases of left-branching and right-branching can be processed with finite memory (finite-state machines), unlike center-embedding which requires unbounded memory (a

[13] http://en.wikipedia.org/wiki/Formal_grammar mentions applications in theoretical computer science, theoretical linguistics, formal semantics and mathematical logic. This article mentions Knuth and ALGOL 68 among many other Computer Science connections. The ALGOL specification defined the syntax of the language in terms of BNF (Bacus-Naur Form), a context-free formalism. This Wikipedia article has more references in Computer Science than Linguists.

[14] http://www-cs-faculty.stanford.edu/~knuth/cl.html

stack).

A simple example of center-embedding is a parenthesis grammar:

$$< expr > \rightarrow (< expr >)$$

Parenthesis grammars are a special case of center-embedding where x is an open parenthesis and y is a closed parenthesis. A stack can easily keep track of the long-distance dependencies between open and close parentheses, but that requires unbounded memory. The big question is whether a parenthesis grammar could be processed with finite memory. Chomsky proved that cannot be done. More generally, finite-state methods cannot capture center-embedding.

Chomsky (1956) used the following examples to argue that English is center-embedded, and therefore, beyond the capabilities of finite-state methods such as HMMs. Chomsky assumed that English has a non-terminal S (for sentence or clause) that generates itself with non-empty material on both sides, as in:[15]

1. $S \rightarrow \cdots \rightarrow$ If S, then S.
2. $S \rightarrow \cdots \rightarrow$ Either S, or S.
3. $S \rightarrow \cdots \rightarrow$ The man who said that S, is arriving today.

Thus far, approximations such as ngrams and finite-state have served us well. While there are obvious limitations with these approximations, so far, it is hard to point to more effective alternatives. Attempts to capture unusual long-distance dependencies tend to fix a few unusual fringe cases, but break more cases than they fix. Engineers have found that it is more important to address more common short-distance dependencies than less common long-distance dependencies. At least, this has been the experience of our generation.

That said, we ought to prepare the next generation for the possibility that they might do better than we have. We ought to teach the next generation about the strengths and weaknesses of currently popular methods. They need to know about the most successful approximations that we know of, but they also need to know about their limitations. It is likely that the next generation will find improvements over ngrams, and they might even find improvements that go beyond finite-state.

1.3.2 Minsky's Objections

Minsky and Papert (1969) showed that perceptrons (and more generally, linear separators) cannot learn functions that are not linearly

[15] There has always been some debate over the center-embedding facts. It is hard to find corpus evidence for more than two or three levels of center-embedded. See http://en.wikipedia.org/wiki/Center_embedding and Church (1980), and references therein.

separable such as XOR and connectedness. In two dimensions, a scatter plot is linearly separable when a line can separate the points with positive labels from the points with negative labels. More generally, in n dimensions, points are linearly separable when there is a $n-1$ dimensional hyperplane that separates the positive labels from the negative labels.

Discrimination Tasks

The objection to perceptrons has implications for many popular machine learning methods including linear regression, logistic regression, SVMs and Naive Bayes. The objection also has implications for popular techniques in Information Retrieval such as the Vector Space Model and Probabilistic Retrieval, as well as the use of similar methods for other pattern matching tasks such as:

1. Word-Sense Disambiguation (WSD): distinguish "river" bank from "money" bank.
2. Author Identification: distinguish the Federalist Papers written by Madison from those written by Hamilton.
3. Information Retrieval (IR): distinguish documents that are relevant to a query from those that are not.
4. Sentiment Analysis: distinguish reviews that are positive from reviews that are negative.

Machine Learning methods such as Naive Bayes are often used to address these problems. Mosteller and Wallace (1964), for example, started with the Federalist Papers, a collection of 85 essays, written by Madison, Hamilton and Jay. The authorship has been fairly well established for the bulk of these essays, but there is some dispute over the authorhship for a dozen. The bulk of the essays are used as a training set to fit a model which is then applied to the disputed documents. At training time, Mosteller and Wallace estimated a likelihood ratio for each word in the vocabulary: $Pr(word|Madison)/Pr(word|Hamilton)$. Then the disputed essays are scored by multiplying these ratios for each word in the disputed essays. The other tasks use pretty much the same mathematics, as illustrated in Table 2.

More recently, discriminative methods such as logistic regression have been displacing generative methods such as Naive Bayes. The objections to perceptrons apply to many variations of these methods including both discriminative and generative variants.

Stoplists, Term Weighting and Learning to Rank

Although the mathematics is similar across the four tasks in Table 2, there is an important difference in stop lists. Information Retrieval

TABLE 2 Four applications of Naive Bayes.

Word Sense Disambiguation (WSD)	$score(context) = \prod_{word\ in\ context} \frac{Pr(word\|sense_1)}{Pr(word\|sense_2)}$
Author Identification	$score(doc) = \prod_{word\ in\ doc} \frac{Pr(word\|author_1)}{Pr(word\|author_2)}$
Information Retrieval (IR)	$score(doc) = \prod_{word\ in\ doc} \frac{Pr(word\|relevant)}{Pr(word\|irrelevant)}$
Sentiment Analysis	$score(doc) = \prod_{word\ in\ doc} \frac{Pr(word\|positive\ review)}{Pr(word\|negative\ review)}$

tends to be most interested in content words, and therefore, it is common practice to use a stop list to ignore function words such as "the." In contrast, Author Identification places content words on a stop list, because this task is more interested in style than content.

The literature has quite a bit of discussion on term weighting. Term weighting can be viewed as a generalization of stop lists. In modern web search engines, it is common to use modern machine learning methods to learn optimal weights. Learning to rank methods can take advantage of many features. In addition to document features that model what the authors are writing, these methods can also take advantage of features based on user logs that model what the users are reading. User logs (and especially click logs) tend to be even more informative than documents because the web tends to have more readers than writers. Search engines can add value by helping users discover the wisdom of the crowd. Users want to know what's hot (where other users like you are clicking). Learning to rank is a pragmatic approach that uses relatively simple machine learning and pattern matching techniques to finesse problems that might otherwise require AI-Complete Understanding.

Here is a discussion on learning to rank from a recent blog:[16]

> Rather than trying to get computers to understand the content and whether it is useful, we watch people who read the content and look

[16] http://glinden.blogspot.com/2007/09/actively-learning-to-rank.html

at whether they found it useful.

> People are great at reading web pages and figuring out which ones are useful to them. Computers are bad at that. But, people do not have time to compile all the pages they found useful and share that information with billions of others. Computers are great at that. Let computers be computers and people be people. Crowds find the wisdom on the web. Computers surface that wisdom.

1.3.3 Why Current Technology Ignores Predicates

Weighting systems for Information Retrieval and Sentiment Analysis tend to focus on rigid designators (e.g., nouns) and ignore predicates (verbs, adjectives and adverbs) and intensifiers (e.g., "very") and loaded terms (e.g., "Mickey Mouse" and "Rinky Dink"). The reason might be related to Minsky and Papert's criticism of perceptrons. Years ago, we had access to MIMS, a collection of text comments collected by AT&T operators. Some of the comments were labeled by annotators as positive, negative or neutral. Rigid designators (typically nouns) tend to be strongly associated with one class or another, but there were quite a few loaded terms were either positive or negative, but rarely neutral.

How can loaded terms be positive? It turns out that the judges labeled the document as good for us if the loaded term was predicated of the competition, and bad if it was predicated of us. In other words, there is an XOR dependency (loaded term XOR us) that is beyond the capabilities of a linear separator.

Current practice in Sentiment Analysis and Information Retrieval does not model modifiers (predicate-argument relationships, intensifiers and loaded terms), because it is hard to make sense of modifiers unless you know what they are modifying. Ignoring loaded terms and intensifiers seems like a missed opportunity, especially for Sentiment Analysis, since they are obviously expressing strong opinions. But you can't do much with a feature if you don't know the sign, even if you know the magnitude is large.

When predicate-argument relationships are eventually modeled, it will be necessary to revisit the linearly separable assumption because of the XOR problem mentioned above.

1.3.4 Pierce's Objections

There is less coverage of Pierce in contemporary textbooks than Minsky and Chomsky, despite the impact that Pierce has had on our field as the chair of the ALPAC committee and the author of "Whither Speech Recognition?" It is not clear why modern textbooks have so little to say about Pierce, given how influential his work has been, both in terms of

terminating funding, as well as citations. It may be that his criticisms are even more inconvenient than Minsky's and Chomsky's. Many have tried to respond to Pierce, but few of the responses are as effective or as worth reading as the original criticisms.

Among Pierce's many accomplishments, he developed PCM (Pulse Code Modulation), a method of coding speech that is closely related to today's WAVE file format, a popular format for storing audio on PCs.[17] In addition, Pierce did significant research on vacuum tubes, but soon brought about their demise by supervising the team that invented the transistor. Pierce worked on satellite research, and later, as Vice President of Research at Bell Labs, Pierce played a major role in transferring satellite technology from research to commercial practice with the development of Telstar 1, the first commercial use of satellites in telecommunications.

In short, Pierce was a highly accomplished executive at the top of his game. The poor folks on the other side of the debate were simply no match. Some of Pierce's debating opponents included junior faculty about to be denied tenure. It wasn't a fair fight. But even so, that is no reason to ignore his contributions to the field, inconvenient as they may be.

Both the ALPAC report and "Whither Speech Recognition" are well worth reading. The ALPAC report is easier to find on the web,[18] but considerably longer. If the reader has limited time, she would be well advised to start with "Whither Speech Recognition" because it is short and crisp and to the point. There are basically two objections in this two-page letter:

1. Evaluation: Pierce objects to evaluation by demos, as well as the kinds of evaluations that are popular today. "It is hard to gauge the success of an attempt at speech recognition even when statistics are given. In general... 95% correct can be achieved for... when... Performance has gone down drastically as.... It is not easy to see a practical, economically sound application for speech recognition with this capability."

2. Pattern matching: Pierce objects to the kind of pattern matching that is common today (e.g., machine learning and speech recognition) as artful deception that is "apt to succeed better and more quickly than science."

[17] WAVE has become synonymous with the term "raw digital audio," according to http://www.codeguru.com/cpp/g-m/multimedia/audio/article.php/c8935/.

[18] http://books.nap.edu/html/alpac_lm/ARC000005.pdf

Criticism of Pattern Recognition

Pierce followed up the "artful deception" remark with a reference to Weizenbaum's doctor program, ELIZA, as an example of such a deception. It might be possible for ELIZA to pass the Turing Test, though obviously ELIZA isn't "intelligent." The ELIZA criticism has since become a standard objection to programs that appear to work better than they do. Here is a definition of the ELIZA effect from Wikipedia.[19]

> The ELIZA effect, in computer science, is the tendency to unconsciously assume computer behaviors are analogous to human behaviors. In its specific form, the ELIZA effect refers only to "the susceptibility of people to read far more understanding than is warranted into strings of symbols—especially words—strung together by computers." More generally, the ELIZA effect describes any situation where, based solely on a system's output, users perceive computer systems as having "intrinsic qualities and abilities which the software controlling the (output) cannot possibly achieve" or "assume that (outputs) reflect a greater causality than they actually do." In both its specific and general forms, the ELIZA effect is notable for occurring even when users of the system are aware of the determinate nature of output produced by the system. From a psychological standpoint, the ELIZA effect is the result of a subtle cognitive dissonance between the user's awareness of programming limitations and their behavior towards the output of the program. The discovery of the ELIZA effect was an important development in artificial intelligence, demonstrating the principle of using social engineering rather than explicit programming to pass a Turing test.

Weizenbaum himself became a strong opponent of AI when he realized just how convincing his ELIZA program was to the public. The following was taken from a chapter of his book titled "Incomprehensible Programs."[20]

> These two programs [MACSYMA and DENDRAL] are distinguished from most other artificial-intelligence programs precisely in that they rest solidly on deep theories. . . . There are, of course, many other important and successful applications of computers. Computers, for example, control entire petroleum-refining plants, navigate spaceships, and monitor and largely control the environments in which astronauts perform their duties. Their programs rest on mathematical control theory and on firmly established physical theories. Such theory-based programs enjoy the enormously important advantage that, when they misbehave, their human monitors can detect that their performance does not correspond to the dictates of their theory and can diagnose

[19] http://joshgreenberg.name/post/153115039/wikipedia-eliza-effect
[20] Weizenbaum (1976, pp. 231–232)

> the reason for the failure from the theory.
>
> But most existing programs... are not theory-based.... They are heuristic... stratagems that appear to "work" under most foreseen circumstances.... My own program, ELIZA, was precisely this type. So is Winograd's Language-understanding system and... Newell and Simon's GPS.[21]

Weizenbaum continues by arguing that programs should be comprehensible, and should be based on solid theoretical foundations, a perspective that Pierce would also agree with.

Pierce's "artful deception" remark is a criticism of proof-by-demos in Artifical Intelligence, as well as speech recognition and the larger area of pattern recognition (and much of modern machine learning).[22]

> Any application of the foregoing discussion to work in the general area of pattern recognition is left as an exercise for the reader.

Pattern recognition has its strengths and weaknesses. On the positive side, pattern recognition makes it possible to make progress on applications by finessing many hard scientific questions. On the other hand, pattern recognition makes it hard to make progress on the key scientific questions because short-term finesses distracts long-term science.

Many engineering tasks share the experience of Speech Synthesis where there have been two threads of research: a pragmatic engineering approach (e.g., concatenative synthesis and tape splicing) and a more ambitious scientific program (e.g., articulatory synthesis). In general, while pragmatic approaches are more likely to produce better results in the short-term, there is considerable sympathy for the more ambitious approach. We have a better chance of making progress on big open scientific questions if we address them head-on rather than finessing around them. That said, if one is in the business of building speech synthesis products, one would be well advised to do whatever finesses it takes to get a quality product out the door on time and on budget.

Responses

There have been many responses to "Whither Speech Recognition," but most responses fail to address the two main criticisms mentioned above:

1. What is the significance of the kinds of evaluations that are required for publication these days?
2. What is the significance of pattern matching (versus science)?

[21] See http://en.wikipedia.org/wiki/General_Problem_Solver for more on GPS.
[22] Pierce (1969, p. 1050)

Roe and Wilpon (1993) argue that the field evolved over the 25 years after "Whither Speech Recognition" from a "futile" endeavor to a commercial reality. They start with a tutorial of popular methods such as Hidden Markov Models (HMMs), a pattern matching technique of the kind that Pierce is objecting to. The tutorial is followed by an evaluation of the kind that is expected these days. The evaluation is intended to demonstrate that the pattern matching techniques are effective, but the evaluation comes to the same kind of conclusion that Pierce characterizes as "hard to gauge."[23]

> In the laboratory, speech recognizers are quite accurate in acoustic pattern matching. In "real-world" conditions, the error rate is much higher.

ALPAC

The considerably longer ALPAC report raises many more objections, many of which are both inconvenient and hard to respond to. The conclusions lead with some good news:[24]

> Today there are linguistic theoreticians who take no interest in empirical studies or in computation. There are also empirical linguists who are not excited by the theoretical advances of the decade–or by computers. But more linguists than ever before are attempting to bring subtler theories into confrontation with richer bodies of data, and virtually all of them, in every country, are eager for computational support. The life's work of a generation ago (a concordance, a glossary, a superficial grammar) is the first small step of today, accomplished in a few weeks (next year, in a few days), the first of 10,000 steps toward an understanding of natural language as the vehicle of human communication.

But the good news is quickly followed by some not-so-good news:[25]

> But, we do not yet have good, easily used, commonly known methods for having computers deal with language data.

Steedman (2008) responds by comparing our field to physics. Steedman observes physics isn't plagued with reports like ALPAC: "Nobody goes around telling physicists what to do." Steedman suggests our field would be in better shape if we were more disciplined, and refrained from airing dirty laundry in public.

We shouldn't dismiss ALPAC with physics envy. That response not only fails to address the issues, but in fact, physics is hardly in an enviable position. There was a time when physics was in relatively good shape, but that was a long time ago. The winter has gone on for so

[23] Roe and Wilpon (1993), p. 58
[24] Pierce et al. (1966), p. 30
[25] Pierce et al. (1966), p. 30

long that many have left the field. Former physicists have contributed to many fields, including fields of interest to our community such as machine translation and machine learning. As for dirty laundry, physics has more than its share.[26]

Even the ALPAC report points out that computational linguistics has a number of advantages over physics:[27]

> We see that the computer has opened up to linguists a host of challenges, partial insights, and potentialities. We believe these can be aptly compared with the challenges, problems, and insights of particle physics. Certainly, language is second to no phenomenon in importance. And the tools of computational linguistics are considerably less costly than the multibillion-volt accelerators of particle physics.

Hutchins (1996)[28] recognized the 30th anniversary of the ALPAC report with a summary article in the MT News International titled: "ALPAC: the (in)famous report." Hutchins concludes (with British spelling):

> ALPAC was quite right to be sceptical about MT: the quality was undoubtedly poor, and did not appear to justify the level of financial support it had been receiving. It was also correct to identify the need to develop machine aids for translators, and to emphasise the need for more basic research in computational linguistics. However, it can be faulted for...

Hutchins continues to criticize the report for taking an excessively American-centric perspective on a question that should be considered in a more global context. Given the seriousness of the lead, the American-centric criticism is relatively minor. If the technology was no good and overpriced from an American perspective, is there another perspective where it would be appropriate for someone else?

In fact, the ALPAC report is remembered as infamous because the skepticism led to a funding winter, especially in the American context. However, the report (page 34) actually recommended expenditures in two distinct areas:

1. Basic long-term academic research in linguistics and computational linguistics, as well as
2. Practical short-term applied work to improve translation practice.

Proposals in the first area should be evaluated by peer review on the basis of scientific merit, whereas the studies in the second area should

[26] See http://www.thetroublewithphysics.com for an example for criticism of physics.

[27] Pierce et al. (1966), p. 30

[28] http://www.hutchinsweb.me.uk/ALPAC-1996.pdf

be evaluated in terms of practical metrics: speed, cost and quality.

These two recommendations call out two sides of Pierce that enable Pierce to endorse two positions as different as Chomsky and Shannon. On the one hand, Pierce was a strong supporter of basic science. Pierce objects to attempts to sell science as something other than it is (e.g., applications), as well as attempts to misrepresent progress with misleading demos and/or mindless metrics (such as the kinds of evaluations that are routinely performed today). On the other hand, there is also a practical side to Pierce, as demonstrated by his impressive accomplishments in speech coding, vacuum tubes, transistors and communication satellites. He is a strong supporter of applied work, but under very different rules, e.g., in terms of a business case. Applied work should be evaluated as applied work (based on a business case), and science should be evaluated as science (based on peer review).

If Pierce were alive today, he would be deeply troubled by the current state of the science, which is heavily invested in pattern matching techniques and numerical evaluations in ways that distract the field from what he would consider to be the core scientific questions.

On a more positive note, the applied side of Pierce would be impressed by Google's business success, especially in search. That said, the success is less clear cut for Google's side businesses in speech recognition and machine translation. While there are some reasons to remain hopeful, a skeptic like Pierce would find it hard to justify the R&D investments the community has made over the decades. For a reasonable return on investment, by now the speech recognition and machine translation community should have produced a killer app, something that almost everyone would use almost every day like AT&T's telephone, or Microsoft Windows or Google Search. Google's core business in search has achieved this bar, and someday their side businesses in speech and translation may eventually do so as well.

What does Pierce have to offer us today? Thus far, the field has done well by picking low-hanging fruit. In good times, when there are lots of easy pickings, we should take advantage of the opportunities. But if those opportunities should dry up, we would be better off following Pierce's advice. It is better to address the core scientific challenges than to continue to look for easy pickings that are no longer there.

1.3.5 Those Who Ignore History Are Doomed To Repeat It

For the most part, the empirical revivals in Machine Learning, Information Retrieval and Speech Recognition have simply ignored PCM's arguments, though in the case of neural nets, the addition of hidden layers to perceptrons could be viewed as a concession to Minsky and Papert.

Despite such concessions, Minsky and Papert (1988) expressed disappointment with the lack of progress since Minsky and Papert (1969).

> In preparing this edition we were tempted to "bring those theories up to date." But when we found that little of significance had changed since 1969, when the book was first published, we concluded that it would be more useful to keep the original text... and add an epilogue.... One reason why progress has been so slow in this field is that researchers unfamiliar with its history have continued to make many of the same mistakes that others have made before them. Some readers may be shocked to hear it said that little of significance has happened in the field. Have not perceptron-like networks–under the new name connectionism–become a major subject of discussion.... Certainly, yes, in that there is a great deal of interest and discussion. Possibly yes, in the sense that discoveries have been made that may, in time, turn out to be of fundamental importance. But certainly no, in that there has been little clear-cut change in the conceptual basis of the field. The issues that give rise to excitement today seem much the same as those that were responsible for previous rounds of excitement.... Our position remains what it was when we wrote the book: We believe this realm of work to be immensely important and rich, but we expect its growth to require a degree of critical analysis that its more romantic advocates have always been reluctant to pursue–perhaps because the spirit of connectionism seems itself to go somewhat against the grain of analytic rigor.[29]

> Multilayer networks will be no more able to recognize connectedness than are perceptrons.[30]

Gaps in Courses on Computational Linguistics

Part of the reason why we keep making the same mistakes, as Minsky and Papert mentioned above, has to do with teaching. One side of the debate is written out of the textbooks and forgotten, only to be revived/reinvented by the next generation. Contemporary textbooks in computational linguistics have remarkably little to say about PCM. Pierce isn't mentioned in Jurafsky and Martin (2000), Manning and Schütze (1999) or Manning et al. (2008). Minsky's criticism of Perceptrons is briefly mentioned in just one of the three textbooks: Manning and Schütze (1999, p. 603). A student new to the field might not appreciate that the reference to "related learning algorithms" (see bold italics below) includes a number of methods that are currently very popular such as linear and logistic regression.

> There are similar convergence theorems for some other gradient descent

[29] Minsky and Papert (1988, Prologue, p. vii)
[30] Minsky and Papert (1988, Epilogue, p. 252)

> algorithms, but in most cases convergence will only be to a local optimum.... Perceptrons converge to a global optimum because they select a classifier from a class of simpler models, the linear separators. There are many important problems that are not linearly separable, the most famous being the XOR problem.... A decision tree can learn such a problem whereas a perceptron cannot. After some initial enthusiasm about Perceptrons (Rosenblatt, 1962), researchers realized these limitations. As a consequence, interest in perceptrons and ***related learning algorithms***[emphasis added] faded quickly and remained low for decades. The publication of Minsky and Papert (1969) is often seen as the point at which the interest in this genre of learning algorithms started to wane.

Manning et al. (2008) have a brief reference to Minsky and Papert (1988) as a good description of perceptrons, with no mention of the sharp criticism.

> Readers interested in algorithms mentioned, but not described in this chapter, may wish to consult Bishop (2006) for neural networks, Hastie et al. (2001) for linear and logistic regression, and Minsky and Papert (1988) for the perceptron algorithm.[31]

Based on this description, a student might come away with the mistaken impression that Minsky and Papert are fans of perceptrons (and currently popular related methods such as linear and logistic regression).

Bishop (2006, p. 193) makes it clear that Minsky and Papert are no fans of perceptrons and neural networks, but dismisses their work as "incorrect conjecture." Bishop points to widespread use of neural networks in practical application as counter-evidence to Minsky and Papert's claim above that "not much has changed" and "multilayer networks will be no more able to recognize connectedness than are perceptrons."

Contemporary textbooks ought to teach both the strengths and the weaknesses of useful approximations such as neural networks. Both sides of the debate have much to offer. We do the next generation a disservice when we dismiss one side or the other with harsh words like "incorrect conjecture" and "not much has changed."

Chomsky receives more coverage than Pierce and Minsky in contemporary textbooks. There are 10 references to Chomsky in the index of Manning and Schütze (1999) and 27 in the index of Jurafsky and Martin (2000). The coverage of Chomsky's criticism of finite-state methods is similar in both textbooks, though the second textbook has more ref-

[31] Manning et al. (2008, p. 292)

erences to Chomsky's work on other topics such as phonology Chomsky and Halle (1968), whereas the first book is focused on Statistical Natural Language Processing, and does not attempt to cover other topics such as Speech.

Both textbooks mention Chomsky's criticism of finite-state methods and the devastating effect that they had on empirical methods at the time, though they quickly move on to describe the revival of such methods, with relatively little discussion of the argument, motivations for the revival, and implications for current practice and the future.

> In a series of extremely influential papers starting with Chomsky (1956) and including Chomsky (1957) and Miller and Chomsky (1963), Noam Chomsky argued that "finite-state Markov processes," while a possibly useful engineering heuristic, were incapable of being a complete cognitive model of human grammatical knowledge. These arguments led many linguists and computational linguists away from statistical models altogether.
>
> The resurgence of N-gram models came from Jelinek, Mercer, Bahl. . . [32]

Both books also start the ngram discussion with a few quotes, pro and con.[33]

> But it must be recognized that the notion "probability of a sentence" is an entirely useless one, under any known interpretation of this term.[34]
>
> Anytime a linguist leaves the group the recognition rate goes up.[35]

Manning and Schütze (1999, p. 2) starts the discussion with these quotes:

> Statistical considerations are essential to an understanding of the operation and development of languages.[36]
>
> One's ability to produce and recognize grammatical utterances is not based on notions of statistical approximations and the like.[37]

Such quotes introduce the student to the existence of a controversy, but they don't help the student appreciate what it means for them. We should remind students that Chomsky objected to a number of finite-state methods that are extremely popular today including ngrams and Hidden Markov Models because such methods cannot capture long-distance dependences (e.g., agreement constraints and wh-movement).

[32] Jurafsky and Martin (2000, pp. 230–231)
[33] Jurafsky and Martin (2000, p. 191)
[34] Chomsky (1965, p. 57)
[35] Fred Jelinek (then of IBM speech group) (1988)
[36] Lyons (1968, p. 98)
[37] Chomsky (1957, p. 16)

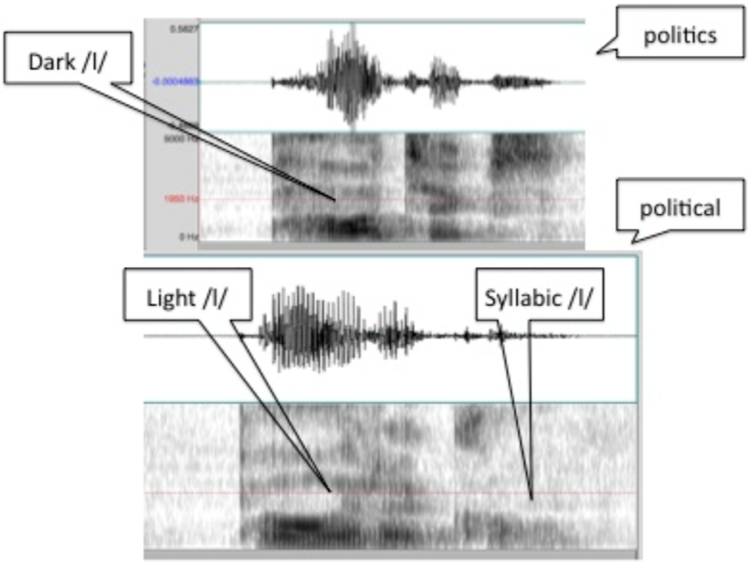

FIGURE 3 The spectrograms of "politics" and "political" show three allophones of /l/. Different allophones appear before and after stress.

Educating Computational Linguistics Students in General Linguistics and Phonetics

To prepare students for what might come after the low hanging fruit has been picked over, it would be good to provide today's students with a broad education that makes room for many topics in Linguistics such as syntax, morphology, phonology, phonetics, historical linguistics and language universals. We are graduating Computational Linguistics students these days that have very deep knowledge of one particular narrow sub-area (such as machine learning and statistical machine translation) but may not have heard of Greenberg's Universals, Raising, Equi, quantifier scope, gapping, island constraints and so on. We should make sure that students working on co-reference know about c-command and disjoint reference. When students present a paper at a Computational Linguistics conference, they should be expected to know the standard treatment of the topic in Formal Linguistics.

Students working on speech recognition need to know about lexical stress (e.g., Chomsky and Halle (1968)). Phonological stress has all sorts of consequences on downstream phonetic and acoustic processes. Speech recognizers currently don't do much with lexical stress which seems like a missed opportunity since stress is one of the more salient

properties in the speech signal. Figure 3 shows waveforms and spectrograms for the minimal pair: "politics" and "political." There are many differences between these two words. The technology currently focuses on differences at the segmental level:

1. "Politics" ends with *-s* where as "political" ends with *-al*.
2. The first vowel in "political" is a reduced schwa unlike the first vowel in "politics."

The differences in stress are even more salient. Among the many stress-related differences, Figure 3 calls out the differences between pre-stress and post-stress allophones of /l/. There are also consequences in the /t/s; /t/ is aspirated in "politics" and flapped in "political."

Currently, there is still plenty of low-hanging fruit to work on at the segmental level, but eventually the state of the art will get past those bottlenecks. We ought to teach students in speech recognition about the phonology and acoustic-phonetics of lexical stress, so they will be ready when the state of the art advances past the current bottlenecks at the segmental level. Since there are long-distance dependencies associated with stress that span over more than tri-phones, progress on stress will require a solid understanding of the strengths and weaknesses of currently popular approximations. Fundamental advances in speech recognition, such as effective use of stress, will likely require fundamental advances to the technology.

1.4 Conclusions

Pierce, Chomsky and Minsky challenged a number of empirical methods that were popular at the time, and have since been revived. Their objections have implications for many popular contemporary methods including Machine Learning (Linear Separators), Information Retrieval (Vector Space Model), Language Modeling (ngrams) and Speech Recognition (Hidden Markov Models (HMMs) and Conditional Random Fields (CRFs)).

Students need to learn how to use popular approximations effectively. Chomsky wrote about limitations with ngrams and Minsky wrote about limitation with linear separators. Many of these limitations are obvious (by construction), but even so, the debate, both pro and con, has been heated at times. And sometimes, one side of the debate is written out of the textbooks and forgotten, only to be revived/reinvented by the next generation. We should encourage the next generation to learn the arguments on both sides of these debates, even if they choose to take one side or the other.

When we revived empiricism in the 1990s, we chose to reject the po-

sition of our teachers for pragmatic reasons. Data had become available like never before. What could we do with it? We argued that it is better to do something simple than nothing at all. Let's go pick some low hanging fruit. While trigrams cannot capture everything, they often work better than the alternatives. It is better to capture the agreement facts that we can capture easily, than to try for more and end up with less.

That argument made a lot of sense in the 1990s, especially given unrealistic expectations that had been raised during the previous boom. But today's students might be faced with a very different set of challenges in the not-too-distant future. What should they do when most of the low hanging fruit has been pretty much picked over?

In the particular case of Machine Translation, the revival of statistical approaches (e.g., Brown et al. (1993)) started out with finite-state methods for pragmatic reasons, but gradually over time, researchers have become more and more receptive to the use of syntax to capture long-distance dependences, especially when there isn't very much parallel corpora, and for language pairs with very different word orders (e.g., translating between a subject-verb-object (SVO) language like English and a verb final language like Japanese). Going forward, we should expect Machine Translation research to make more and more use of richer and richer linguistic representations. So too, there will soon be a day when stress will become important for speech recognition.

Since it isn't possible for textbooks in computational linguistics to cover all of these topics, we should work with colleagues in other departments to make sure that students receive an education that is broad enough to prepare them for all possible futures, or at least all probable futures.

References

Bishop, Christopher. 2006. *Pattern Recognition and Machine Learning*. New York: Springer.

Brown, Peter, John Cocke, Stephen Pietra, Vincent Pietra, Frederick Jelinek, Robert Mercer, and Paul Roossin. 1988. A statistical approach to language translation. In *COLING*.

Brown, Peter, Vincent Pietra, Stephen Pietra, and Robert Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics* 19:263–311.

Chomsky, Noam. 1956. Three models for the description of language. In *IRE Transactions on Information Theory*, vol. 2, pages 113–124.

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

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

Chomsky, Noam and Morris Halle. 1968. *The Sound Pattern of English*. New York: Harper & Row.

Church, Kenneth. 1980. On memory limitations in natural language processing. Tech. Rep. MIT/LCS/TR-245, MIT.

Church, Kenneth. 1988. A stochastic parts program and noun phrase parser for unrestricted text. In *In Proceedings of the Second Conference on Applied Natural Language Processing*, pages 136–143.

Church, Kenneth and Robert Mercer. 1993. Introduction to the special issue on computational linguistics using large corpora. *Computational Linguistics* 19:1–24.

Firth, John Rupert. 1957. A synopsis of linguistic theory 1930-1955. In *Special Volume of the Philological Society*. Oxford: Oxford University Press.

Francis, W. Nelson and Henry Kucera. 1982. *Frequency Analysis of English Usage*. Boston: Houghton Mifflin.

Hall, David, Daniel Jurafsky, and Christopher Manning. 2008. Studying the History of Ideas Using Topic Models. In *EMNLP*, pages 363–371.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. 2001. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag.

Hutchins, John. 1996. ALPAC: The (In)famous report. In *MT News International*, pages 9–12.

Jurafsky, Daniel and James Martin. 2000. *Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics*. Prentice-Hall.

Lyons, John. 1968. *Introduction to theoretical linguistics*. Cambridge, England: Cambridge University Press.

Manning, Christopher, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press. ISBN 0521865719.

Manning, Christopher and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. Cambridge, MA: MIT Press.

Miller, George and Noam Chomsky. 1963. Finitary Models of Language Users. In D. Luce, R. Bush, and E. Galanter, eds., *Handbook of Mathematical Psychology*, vol. 2, pages 419–491. New York: Wiley.

Minsky, Marvin and Seymour Papert. 1969. *Perceptrons*. Cambridge, MA: MIT Press.

Minsky, Marvin and Seymour Papert. 1988. *Perceptrons*. Cambridge, MA: MIT Press.

Pierce, John. 1961. *An Introduction to Information Theory: Symbols, Signals and Noise*. New York: Dover Publications, Inc.

Pierce, John. 1969. Whither Speech Recognition. *Journal of the Acoustical Society of America* 46(4P2):1049–1051.

Pierce, John. 1970. Whither Speech Recognition II. *Journal of the Acoustical Society of America* 47(6B):1616–1617.

Pierce, John, John Carroll, Eric Hamp, David Hays, Charles Hockett, Anthony Oettinger, and Alan Perlis. 1966. *Language and Machines: Computers in Translation and Linguistics*. Washington, D.C.: National Academy of Sciences, National Research Council.

Roe, David and Jay Wilpon. 1993. Whither Speech Recognition: The Next 25 Years. *IEEE Communications* 31(11):54–63.

Rosenblatt, Frank. 1962. *Principles of Neurodynamics; Perceptrons and the Theory of Brain Mechanisms*. Washington: Spartan Books.

Simon, Herb. 1960. Management by machines: How much and how soon? *The Management Review* 49:12–19 and 68–80.

Sinclair, John. 1987. *Looking Up: An Account of the COBUILD Project in Lexical Computing*. Glasgow: Collins.

Sinclair, John, Patrick Hanks, Gwyneth Fox, Rosamund Moon, and Penny Stock, eds. 1987. *Collins COBUILD English Language Dictionary*. Glasgow: Collins.

Steedman, Mark. 2008. On Becoming a Discipline. *Computational Linguistics* 34(1):137–144.

Tukey, John. 1977. *Exploratory Data Analysis*. Reading, MA: Addison-Wesley.

Weizenbaum, Joseph. 1976. *Computer Power and Human Reason*. San Francisco: W. H. Freeman.