in *Beliefs, Reasoning, and Decision Making: Psycho-logic in Honor of Bob Abelson* (eds: Schank & Langer), Lawrence Erlbaum Associates, Hillsdale, NJ. pp. 143-173.

Cognition, Computers, and Car Bombs: How Yale Prepared Me for the 90's

Wendy G. Lehnert
Department of Computer Science
University of Massachusetts
Amherst, MA 01003

From Yeshiva to Yale (1974)

From the East Side in midtown Manhattan, it was a brisk 20-minute walk going west to Eighth Avenue, another 30 minutes going north on the A train to 182nd Street, and then a final 10-minute walk going east to get to the Belfer Graduate School of Science at Yeshiva University. I made that trip every day for two years as a graduate student in mathematics. The math department was housed in a modern high-rise that stood out among the older and less majestic buildings of Washington Heights. Within that seemingly secular structure, each office door frame was uniformly adorned with a small white plastic mezuzah, courtesy of the university.

I thought a lot about what I was doing with my life during those subway rides. It was probably on the subway that I realized I was more interested in how mathematicians manage to invent mathematics than I was in the actual mathematics itself. I mentioned this to one of my professors, and his reaction was polite but pointed. Academic math departments had no room for dilettantes. Anyone who was primarily interested in the cognitive processes of mathematicians did not belong in mathematics and was well-advised to pursue those interests elsewhere. Given the abysmal state of the job market for PhDs in mathematics, I took the hint and set out to broaden my horizons.

One day I was browsing in the McGraw-Hill bookstore, and I stumbled across a collection of early writings on artificial intelligence (Feigenbaum and Feldman 1963). It was here that I learned about a community of people who were trying to unravel the mysteries of human cognition by playing around with computers. This seemed a lot more interesting than Riemannian manifolds and Hausdorff spaces, or maybe I was just getting tired of all that time on the subway. One way or another, I decided to apply to a graduate program in computer science just in case there was some stronger connection between FORTRAN and human cognition than I had previously suspected. When Yale accepted me, I decided to throw all caution to the wind and trust the admissions committee. I packed up my basenji and set out for Yale in the summer of 1974 with a sense of grand adventure. I was moving toward light and truth, and my very first full screen text editor.

As luck would have it, Professor Roger Schank, a specialist in artificial intelligence (AI) from Stanford, was also moving to Yale that same summer. Unlike me, Schank knew quite well what to expect in New Haven. He was moving to Yale so he could collaborate with a famous social psychologist, Robert Abelson, on models of human memory. Within a few short months, a fruitful collaboration between Schank and Abelson was underway. Schank was supporting a group of enthusiastic graduate students, and I was writing LISP code for a computer program that read stories and answered simple questions about those stories. [1]

---

[1] LISP may not be significantly closer to human cognition than FORTRAN, but it does drive home the difference between number crunching and symbol crunching. Mainstream artificial intelligence operates on the assumption that intelligent information processing can be achieved through computational symbol manipulation.

I was amazed to discover how difficult it is to get a computer to understand even the simplest sentences, and I began to think about what it means for a human to understand a sentence. I wasn't particularly interested in the problems of vague or misleading language when what you heard isn't quite the same thing as what was said. I was more preoccupied with seemingly trivial sentences like "John gave Mary a book," and the underlying mechanisms that enable us to understand that giving a book is conceptually different from giving a kiss. Two things about this phenomenon seemed astonishing to me. First, it was remarkable that people ever managed to communicate anything at all with their sentences. And second, there appeared to be no body of expertise that could shed much light on the mental processes associated with this most mundane level of language comprehension.

Another Yale computer scientist, Alan Perlis (famous for his APL one-liners among other things), was rather adept at witty aphorisms. One of my favorites was this one: *With computers, everything is possible and nothing is easy*. While the first claim constitutes an article of faith, the second claim is readily apparent to anyone who has ever written a computer program. I believed without question that computers could be made to understand sentences. Even so, it was humbling to discover that the bland activities of John and Mary were somehow more elusive to me than highly abstract theorems of differential geometry and functional analysis. In fact, I was beginning to suspect that one might possibly devote an entire lifetime to John and Mary and the book without ever getting it quite right. It is true that John and Mary lack the intellectual cachet of high powered mathematics, but I no longer believed that my mathematician friends had a monopoly over all the hard problems.

Fast Forward (June 1991)

The place is the Naval Ocean Systems Center in San Diego. I am attending a relatively small, invitation-only meeting with one of my graduate students. The purpose of the meeting is to discuss the outcome of a rigorous performance evaluation in text extraction technologies. Fifteen laboratories have labored for some number of months (one person/year of effort, on average) to create computer systems that can comprehend news stories about terrorism. Each system has taken a rigorous test designed to assess its comprehension capabilities. This particular test consisted of 100 texts, previously unseen by any of the system developers, which were distributed to each of the participating laboratories along with strict testing procedures. Each system was required to (1) extract a database of essential facts from the texts without any human intervention, and (2) be graded against a hand-coded database containing all the correctly encoded facts. The scoring of the test results was conducted by yet another system (the scoring program) which was scrupulously precise and relentlessly thorough in its evaluations.

This unusual meeting is called MUC-3 (a.k.a. the Third Message Understanding Conference), and three university sites have participated in the evaluation along with 12 industry labs. My student and I represent the University of Massachusetts at Amherst. Most of the people here have been involved with natural language processing for at least a decade or more. We no longer discuss how to tackle "John gave Mary a book." Now we debate different ways to measure recall and precision and overgeneration. We talk about spurious template counts, grey areas in the domain guidelines, and whether or not our training corpus of 1300 sample texts was large enough to provide an adequate training base. When we talk about specific sentences at all, we talk about real ones:

```
THE CAR BOMB WAS LEFT UNDER THE BRIDGE ON 68TH STREET AND 13TH
STREET WHERE IT EXPLODED YESTERDAY  AT APPROXIMATELY 1100,
KILLING MARIA JACINTA PULIDO, 42;  PILAR PULIDO, 19; A MINOR
REPORTEDLY KNOWN AS CARLOS; EFRAIN RINCON RODRIGUEZ, AND A POLICE
OFFICIAL WHO DIED AT THE POLICE CLINIC.
```

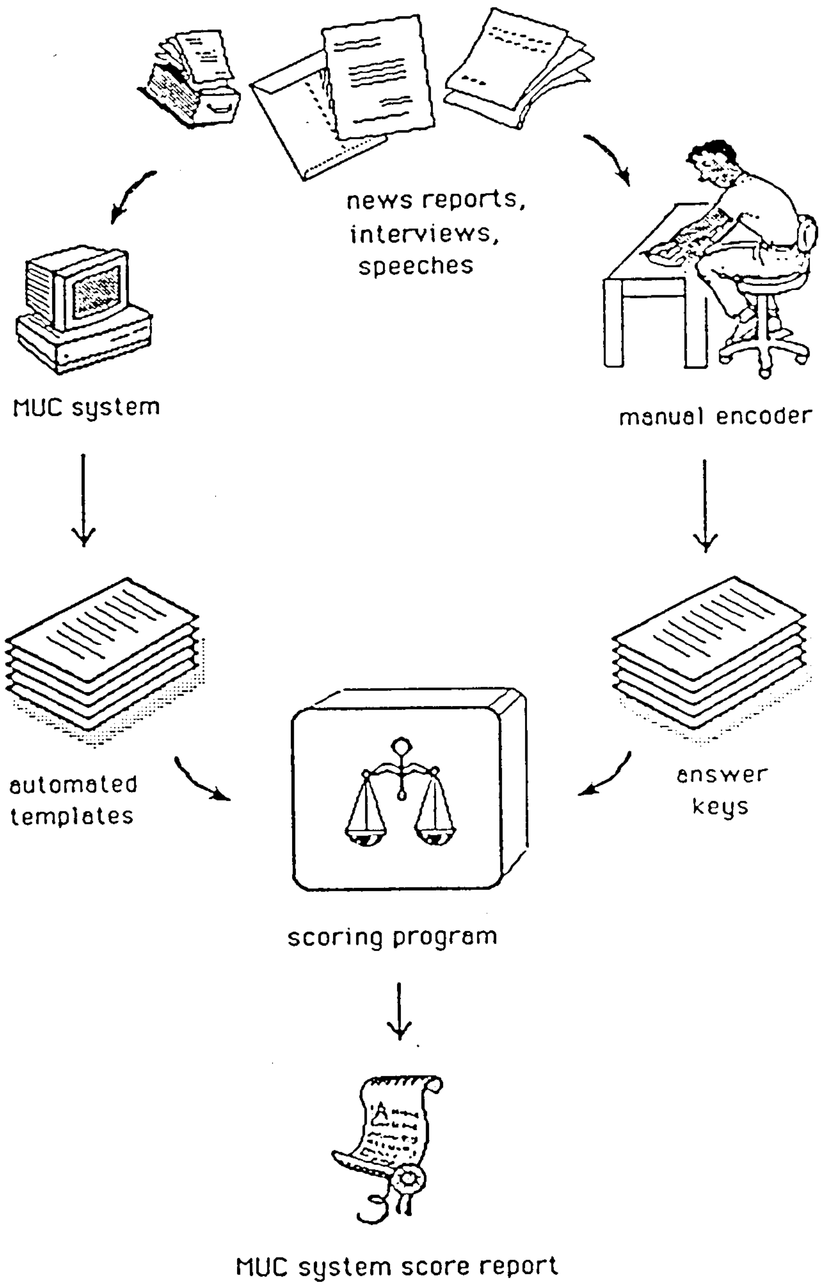

Figure 1: The MUC Method for Evaluating Computational Text Analyzers

The purpose of the meeting is threefold. First, we are hoping to assess the state-of-the-art in natural language processing as it applies to information extraction tasks. Second, we would like to achieve some greater understanding about which approaches work well and which approaches are choking. Third, we are interested in the problem of technology evaluations and what it takes to get an objective assessment of our respective systems. Many of us bring hard-won years of experience and research to the table. We are curious to see what all that background will buy us. MUC-3 is an exciting meeting because it signifies a first attempt at a serious evaluation in natural language processing. Evaluations had been conducted prior to 1991 in speech recognition[2], but nothing has been attempted in natural language processing until now.

Before we discuss the outcome of the evaluation, some observations about the MUC-3 meeting are in order. Most importantly, the researchers who are here represent a broad spectrum of approaches to natural language processing. MUC-3 attracted formal linguists who concentrate on complicated sentence grammars, connectionists who specialize in models of neural networks, defense contractors who happily incorporate any idea that looks like it might work, and skilled academics who have based entire research careers on a fixed set of assumptions about the problem and its solutions. It is unusual to find such an eclectic gathering under one roof. The social dynamics of the MUC-3 meeting are an interesting topic in its own right.

On the one hand, we have 15 sites in apparent competition with one another. On the other hand, we have 15 sites with a strong common bond. Each MUC-3 site knew all too well the trauma of preparing for MUC-3 and the uneasy prospect of offering up the outcome for public scrutiny. Like the survivors of some unspeakable disaster, we have gathered in San Diego to trade war stories, chuckle over the twisted humor that is peculiar to folks who have been spending a little too much time with their machines, and explore a newfound sense of common ground that wasn't there before. We all wanted to understand what made each of the systems work or not work. We all wanted to identify problem areas of high impact. And we all wanted to see where we each stood with respect to competing approaches. It was an intensely stimulating meeting.

Flash Back (1978)

This episode takes place at Tufts University in Boston. Roger Schank and Noam Chomsky have agreed to appear in a piece of academic theater casually known as The Great Debate. On Chomsky's side we have the considerable momentum of an intellectual framework that reshaped American linguistics throughout the 60s: a perspective on language that is theoretical, permeated with abstractions, and exceedingly careful to distinguish the theoretical aspects of language from empirical language phenomena. On Schank's side we have a computational perspective on language that emphasizes human memory models, inference generation, and the claim that meaning is the engine that drives linguistic communication. Chomsky proposes formalisms that address syntactic structures: from this perspective he attempts to delineate the innate nature of linguistic competence. Schank is interested in building systems that work: he makes pragmatic observations about what is needed to endow computers with human-like language facilities.

Chomsky rejects the computational perspective because it resists containment by an agreeable formalism. Schank rejects Chomsky's quest for formalisms because the problem he wants to solve is big and messy and much harder than anything that any of the formalists are willing to tackle. As with all Great Debates, both sides are passionately convinced that the opposition is hopelessly deluded. There is no common ground. There is no room for compromise. There is no resolution.

---

2 *Speech recognition* refers to the comprehension of spoken language, as opposed to *natural language processing* which assumes input in the form of written language.

I won't say who won The Great Debate. I wasn't there myself. But various manifestations of The Great Debate haunted much of Schank's academic life in one way or another throughout much of the 70's. As a graduate student in Schank's lab, I was thoroughly sensitized to a phenomena that is not unrelated to The Great Debate. It was a phenomenon associated with methodological styles. Simply put, some researchers are problem-driven and some researchers are technology-driven.

Problem-driven researchers start with a problem and look for a technology that can handle the problem. Sometimes nothing works very well and a new technology has to be invented. Technology-driven researchers start with a technology and look for a problem that the technology can handle. Sometimes nothing works very well and a new problem has to be invented. Both camps are equally dedicated and passionate about their principal alliance. Some of us fall in love with problems and some of us fall in love with technologies. Does a chicken lay eggs to get more chickens or do eggs make chickens to get more eggs?

As a student who was privileged to attend many research meetings with Bob Abelson, I learned that thought processes and personality traits often interact in predictable ways. Moreover, community standards and social needs are important variables in cognitive modeling. When you turn the lessons of social psychology back on to the scientific community, you discover that researchers, being just as human and social as anyone else, exhibit many predictive features that correlate with specific intellectual orientations. In particular, certain personality traits go hand and hand with certain styles of research. Schank and Abelson hit upon one such phenomenon along these lines and dubbed it the neats vs. the scruffies. These terms moved into the mainstream AI community during the early 80s, shortly after Abelson presented the phenomenon in a keynote address at the Annual Meeting of the Cognitive Science Society in 1981. Here are some selected excerpts from the accompanying paper in the proceedings:

> "The study of the knowledge in a mental system tends toward both naturalism and phenomenology. The mind needs to represent what is out there in the real word, and it needs to manipulate it for particular purposes. But the world is messy, and purposes are manifold. Models of mind, therefore, can become garrulous and intractable as they become more and more realistic. If one's emphasis is on science more than on cognition, however, the canons of hard science dictate a strategy of the isolation of idealized subsystems which can be modeled with elegant productive formalisms. Clarity and precision are highly prized, even at the expense of common sense realism. To caricature this tendency with a phrases from John Tukey (1969), the motto of the narrow hard scientist is, "Be exactly wrong, rather than approximately right".
>
> The one tendency points inside the mind, to see what might be there. The other points outside the mind, to some formal system which can be logically manipulated [Kintsch et al., 1981]. Neither camp grants the other a legitimate claim on cognitive science. One side says, "What you are doing may seem to be science, but it's got nothing to do with cognition." The other side says, "What you're doing may seem to be about cognition, but it's got nothing to do with science."
>
> Superficially, it may seem that the trouble arises primarily because of the two-headed name cognitive science. I well remember discussions of possible names, even though I never liked "cognitive science", the alternatives were worse: abominations like "epistology" or "representonomy".
>
> But in any case, the conflict goes far deeper than the name itself. Indeed, the stylistic division is the same polarization that arises in all fields of science, as well as in art, in politics, in religion, in child rearing -- and in all spheres of human endeavor. Psychologist Silvan Tomkins (1965) characterizes this overriding conflict as that between characterologically left-wing and right-wing world views. The left-wing personality finds the sources of value and truth to lie within individuals, whose reactions to the world define what is important. The right-wing personality

asserts that all human behavior is to be understood and judged according to rules or norms which exist independent of human reaction. A similar distinction has been made by an unnamed but easily guessed colleague of mine, who claims that the major clashes in human affairs are between the "neats" and the "scruffies". The primary concern of the neat is that things should be orderly and predictable while the scruffy seeks the rough-and-tumble of life as it comes ...

The fusion task is not easy. It is hard to neaten up a scruffy or scruffy up a neat. It is difficult to formalize aspects of human thought that which are variable, disorderly, and seemingly irrational, or to build tightly principled models of realistic language processing in messy natural domains. Writings about cognitive science are beginning to show a recognition of the need for world-view unifications, but the signs of strain are clear ...

Linguists, by and large, are farther away from a cognitive science fusion that are the cognitive psychologists. The belief that formal semantic analysis will prove central to the study of human cognition suffers from the touching self-delusion that which is elegant must perforce be true and general. Intense study of quantification and truth conditions because they provide a convenient intersection of logic and language will not prove any more generally informative about the range of potential uses of language than the anthropological analysis of kinship terms told us about culture and language. On top of that, there is the highly restrictive tradition of defining the user of language as a redundant if not defective transducer of the information to be found in the linguistic corpus itself. There is no room in this tradition for the human as inventor and changer and social transmitter of linguistic forms, and of contents to which those forms refer. To try to understand cognition by a formal analysis of language seems to me like trying to understand baseball by an analysis of the physics of what happens when an idealized bat strikes an idealized baseball. One might learn a lot about possible trajectories of the ball, but there is no way in the world one could ever understand what is meant by a double play or a run or an inning, much less the concept of winning the World Series. These are human rule systems invented on top of the structural possibilities of linguistic forms. Once can never infer the rule systems from a study of the forms alone.

Well, now I have stated a strong preference against trying to move leftward from the right. What about the other? What are the difficulties in starting our from the scruffy side and moving toward the neat? The obvious <u>advantage</u> is that one has the option of letting the problem areas itself, rather than the available methodology, guide us about what is important. The obstacle, of course, is that we may not know how to attack the important problems. More likely, we may think we know how to proceed, but other people may find our methods sloppy. We may have to face accusations of being <u>ad hoc</u>, and scientifically unprincipled, and other awful things.

(pp. 1-2 from [Abelson 81])

I periodically go back to this paper, about once every year or two, to think about Abelson's observations in the context of my current research activities. I am always surprised to find new light and truth shining through with each subsequent reading. [3]

The Ad Hoc Thing (1975)

This flashback takes place on the campus of the Massachusetts Institute of Technology. I am giving my first conference talk at TINLP (Theoretical Issues in Natural Language Processing). Schank and Abelson have been promoting the idea of scripts as a human memory structure and my talk describes work with scripts as well [Lehnert 1975]. Minsky's notion of a frame is also getting a lot of attention, and

---

[3] I am not the only one who still thinks about the neats and the scruffies. Marvin Minsky recently published a paper called "Logical Versus Analogical or Symbolic Versus Connectionist or Neat Versus Scruffy" [Minsky 1991].

people seem generally interested in both scripts and frames. The newly formed Yale AI Project is there en masse (Rich Cullingford, Bob Wilensky, Dick Proudfoot, Chris Riesbeck, Jim Meehan and myself). The term "scruffy" hasn't been coined yet but the Yale students have begun to understand that they occupy some position left-of-center in the methodological landscape.

The Yale students had worked very hard in the weeks prior to TINLP completing a system implementation called SAM. SAM was able to read a few exceedingly short stories about someone who went to a restaurant. Afterwards SAM proceeded to answer a small number of questions about the story it had read. SAM used a sentence analyzer, a script application mechanism to create its memory for the story, and procedures for locating answers to questions in memory. Schank was very adamant that SAM had to be running in time for TINLP so that Schank could talk about a *working* computer system that exhibited specific I/O behavior.

SAM was presented as a prototype with serious limitations, designed only to demonstrate a weak approximation to human cognitive capabilities. As such, SAM was lacking in generality at every possible opportunity. SAM at this time only knew about one script (whereas people have hundreds or thousands of them). SAM's vocabulary did not extend beyond the words needed to process its two or three stories. SAM's question answering heuristics had not been tested on questions other than the ones that were presented. SAM's general knowledge about people and physical objects was infinitesimal. In truth, SAM was carefully engineered to handle the input it was designed to handle and produce the output it was designed to produce.

Having said all that, we should explain that SAM was primarily engineered to illustrate the theoretical notion of script application (in the absence of any theoretical foundation, SAM could have been written as a simple exercise in finite table lookup). Even in that respect, corners were cut and simplifying assumptions were made. But despite all the caveats and disclaimers, we were proud that so many of us had been able to coordinate our individual efforts on what was in fact a fairly complicated system by 1975 standards.[4]

There was never any intent to mislead anyone about the limitations of SAM. No one at Yale thought that SAM was a serious system beyond its original intent as a demonstration prototype.[5] We all understood that SAM was exceedingly delicate and generously laced with gaping holes. SAM tossed around references to lobsters and hamburgers, but it didn't really have any knowledge about lobsters and hamburgers beyond a common semantic feature (*FOOD*). It knew that waitresses (*WAITRESS*) bring meals to restaurant patrons (*PATRON*) but it knew nothing about blue collar lifestyles, dead-end jobs, the minimum wage, or life in the food chain. With all of its shortcomings, how could anyone take SAM seriously?

I think the proper answer to this question involves amoebas. Amoebas are lowly life forms and they are bound to disappoint anyone looking for a good conversation. But does that mean the amoebae is a failure? Of course not. Amoebas fall short only if you were expecting something more. SAM was a lot like an amoebae that somehow managed to look like a gifted conversationalist at first glance. As soon as the truth set in, SAM was an inevitable disappointment. It is remarkably easy for people to attribute

---

[4] The Yale DEC-10 time-sharing system could only run the complete SAM system as a stand-alone job. The full core image for SAM occupied about 100k of RAM. I have a personal computer on my desk today with enough RAM to hold 300 copies of the original SAM system in active memory. I may be getting older, but my toys just get better and better.

[5] Later on other script-based system implementations attempted to attain a somewhat more serious status by incorporating multiple scripts and managing greater coverage with respect to various minor details. Richard Cullingford's doctoral dissertation was based on an implementation of SAM that went far beyond the system we were running at the time of this early TINLP meeting. [Cullingford 1978].

intelligence to a computer[6]. It seems childish to blame the computer for the fact that people (even smart people) are sometimes easy marks. But there is yet another aspect to SAM that is both more subtle and more disturbing.

SAM was just a prototype. As such, it didn't have to be robust or bullet proof. It just had to demonstrate the computational viability of scripts as a hypothetical memory structure. If someone really wanted to include all the scriptal knowledge that an average person might have about restaurants, it would take more work than we put into SAM. This was a question of scale, and the scaling problem would probably make a good PhD thesis someday.

In the meantime, it was a good idea to feed SAM stories that were reverse-engineered to stay inside the limitations of SAM's available scripts. To make a whole story run successfully, a lot of reverse-engineering was needed at all possible stages. First, we had to make sure we didn't step outside the boundaries of the available scripts. We deliberately worked to stay within the confines of a very small lexicon in order to simplify the job of the sentence analyzer. We also had to make sure we worded the sentences in a way that would be safe for the limitations of the sentence analyzer. Whatever it was, no one claimed SAM was a robust system. For this reason, SAM was often dismissed as an "ad hoc" system. SAM could only do what SAM was designed to do.[7]

As a graduate student, I was repeatedly reassured that it was necessary to walk before one could hope to run. Robust systems are nothing to worry over when you're still trying to master the business of putting one foot in front of the other. When I became a professor I said the same thing to my own students, but with a growing unease about the fact that none of these robustness issues had budged at all in 5 or 10 or 15 years. At one time I think we all believed robustness was something that would be taken care of by other people, some group of people someplace else who had nothing better to do. For example, there were people in industry who dealt with the D part of R&D. Since professors only address the R part of R&D, it made sense that none of the prototypes built at our universities were robust.[8] Eventually, it became apparent that nobody was dealing with robustness under R or D or anywhere.[9]

---

6 This fact so disturbed Joel Weizenbaum when he saw how people reacted to a computer program called ELIZA, that he began a personal crusade against artificial intelligence. He argued that people would never be capable of looking at a seemingly smart machine and understand that it was really very dumb. [Weizenbaum 1976].

7 When it comes to AI, systems are somehow expected to amaze us by doing something smart that was never anticipated by the programmers. Other areas of computer science do not generally look for this element of surprise.

8 In fact, some would argue that professors shouldn't even work on the R end of R&D. Professors are most often associated with basic research, while the research associated with R&D is more derivative, and largely dependent on basic research. The difference is that research within an R&D framework is always directed toward some hopeful application or product. Basic research, on the other hand, is conducted only to expand the boundaries of human knowledge. Basic research produces knowledge for the sake of knowledge. R&D research produces knowledge from which we expect to derive some concrete benefits. Basic research often fuels R&D efforts in unexpected ways, and must be carefully nurtured without a concern for immediate payoffs. Since we can never know which basic research will eventually pay off, it is foolish to think that basic research can be directed with an eye toward greater productivity .

9 In artificial intelligence, the dichotomy between basic research and practical system development has always been reinforced by an awkward distance between the universities and the commercial sector. Professors and graduate students advance professionally by publishing original results. Their corporate counterparts advance by building working systems. It was perhaps simplistic to assume that the ideas nurtured at a university would readily scale up into working systems once they were moved into industry laboratories. But very few AI people in those days were thinking about the problems of technology transfer. AI was very young and it seemed unreasonable to reach for mature technologies quite so fast.

AI in the 90s: Scaling Up and Shaking Down

By the end of the 80s, a lot of people knew that the robustness problem wasn't going to go away without a concerted effort. Some talk had surfaced about the substantial difficulties of "scaling up" robust natural language applications [Schank 1991]. Scaling up began to assume status as a serious issue worthy of serious attention, and those of us who worked closer to the R part of R&D were encouraged to solve the robustness problem since no one else seemed to be making much headway with it. By the time we got into the 90s, portability and scalability had become legitimate issues for basic research. In particular, our government sponsors were increasingly preoccupied with the question of whether or not certain technologies would scale up. It was within this atmosphere that the series of Message Understanding Conferences were first conceived. The Third Message Understanding Conference (MUC-3) was specifically designed to encourage the construction of large ambitious natural language processing systems. When it comes to issues of scale, there is no substitute for getting right to it and building a large system.

Much can be learned from building large systems. It is an experience that all computer scientists should have, and especially all AI researchers. One lesson that it teaches is the importance of the 80-20 rule. The 80-20 rule applies to many kinds of systems, computational and otherwise. In general, the 80-20 rule says that natural distributions of quantifiable causes and effects are usually pretty skewed. For example, an administrator might discover that 20% of the workforce is responsible for 80% of all medical claims. Or a teacher might be struggling with the fact that 20% of her students take up 80% of her time. The 80-20 rule is as ubiquitous as the curve of diminishing returns and has a lot of implications for shifting cost/benefit ratios.

Although it would be difficult to prove, I would not be at all surprised if someone claimed to have found that 20% of linguistic theory accounts for 80% of actual language phenomena. In fact, I suspect the split might be more like 10% and 90%. By language phenomena, I am referring to actual language use, which automatically renders my hunch irrelevant to the concerns of most practicing linguists.[10] Linguists operate in a paradigm that gives them no good reason to worry about the 80-20 rule.

The 80-20 rule is where theoretical linguistics parts company with computational models of linguistic performance. When you set out to build a big complicated system that simulates human language processing capabilities, it is a very good idea to have some sense of how the 80-20 rule applies to your system. Do 20% of the dictionary definitions account for 80% of your dictionary look ups? Do 20% of the grammar rules account for 80% of the sentences processed? Is 20% of the system responsible for 80% of its errors? Knowledge of these relationships is crucial to efficient system design and development.

In particular, anyone concerned with robustness needs to understand how the 80-20 rule kicks in for very large dictionaries, very large rule bases, and very large input loads. One can never hope to attain complete robustness any more than one can hope to completely master Webster's Dictionary. The idea is to maximize robustness without getting caught up in some arcane and idiosyncratic problem that only pops up once in a hundred years. If 20% of your code is responsible for 80% of your functionality, it's a good idea to identify that 20% and concentrate development efforts there. Robust systems are like good trial lawyers: they may not know everything there is to know, but they tend to know everything they need to know.

---

[10] Chomsky introduced the competence/performance distinction in a deliberate move to separate linguistics from the study of language as it is used by real people engaged in real communication [Chomsky 1965].

MUC-3: Facts and Lessons (1991)

A serious description of the MUC-3 tests and evaluation becomes fairly involved and cannot be covered in the space of a few paragraphs. For our purposes we'll show some sample input, some sample output, and a sample score report. A more comprehensive overview can be found in [Lehnert and Sundheim 1991].

Here is a sample text taken from the TST1 test set:

```
TST1-MUC3-0004

  BOGOTA, 30 AUG 89 (INRAVISION TELEVISION CADENA 2) -- [TEXT] LAST
 NIGHT'S TERRORIST TARGET WAS THE ANTIOQUIA LIQUEUR PLANT. FOUR POWERFUL
 ROCKETS WERE GOING TO EXPLODE VERY CLOSE TO THE TANKS WHERE 300,000
 GALLONS OF THE SO-CALLED CASTILLE CRUDE, USED TO OPERATE THE BOILERS, IS
 STORED. THE WATCHMEN ON DUTY REPORTED THAT AT 2030 THEY SAW A MAN AND A
 WOMAN LEAVING A SMALL SUITCASE NEAR THE FENCE THAT SURROUNDS THE PLANT.
 THE WATCHMEN EXCHANGED FIRE WITH THE TERRORISTS WHO FLED LEAVING BEHIND
 THE EXPLOSIVE MATERIAL THAT ALSO INCLUDED DYNAMITE AND GRENADE ROCKET
 LAUNCHERS, METROPOLITAN POLICE PERSONNEL SPECIALIZING IN EXPLOSIVES,
 DEFUSED THE ROCKETS. SOME 100 PEOPLE WERE WORKING INSIDE THE PLANT.

  THE DAMAGE THE ROCKETS WOULD HAVE CAUSED HAD THEY BEEN ACTIVATED CANNOT
 BE ESTIMATED BECAUSE THE CARIBE SODA FACTORY AND THE GUAYABAL RESIDENTIAL
 AREA WOULD HAVE ALSO BEEN AFFECTED.

  THE ANTIOQUIA LIQUEUR PLANT HAS RECEIVED THREATS IN THE PAST AND MAXIMUM
 SECURITY HAS ALWAYS BEEN PRACTICED IN THE AREA. SECURITY WAS STEPPED UP
 LAST NIGHT AFTER THE INCIDENT. THE LIQUEUR INDUSTRY IS THE LARGEST FOREIGN
 EXCHANGE PRODUCER FOR THE DEPARTMENT.
```

System output is formatted in template structures which look very much like the hand-coded answer keys used to evaluate system performance. Each text may have one answer key, more than one answer key, or no answer keys if the text is deemed to be irrelevant. The text given above has one answer key associated with it (see Figure 2).

The scoring program evaluates overall system performance by checking each output template against the available answer keys. Information must be properly positioned in the right slots and in the right answer keys in order to be counted correct. *Recall* measures the ratio of correct information extracted from the texts against all the available information present in the texts. *Precision* measures the ratio of correct information that was extracted against all the information that was extracted.[11] For a test run of 100 texts, recall and precision is averaged over all the output templates and computed four different ways. Figure 3 shows the official UMass score report from the final MUC-3 test run.

Although four scoring metrics were computed for each score report, MATCHED/MISSING was designated as the official scoring metric for MUC-3. Figure 4 shows a scatter plot for the recall and precision of all 15 sites under the official scoring metric.

---

[11] For example, suppose you answered two of four test questions correctly. Your recall score for the test would be 50%. But your precision score for the test would depend on how many questions you answered altogether. If you answered all four, your precision would be 50%. If you answered three questions, your precision would be 75%, and if you were smart enough to only answer two questions, then your precision would be 100%.

It is easy to drown in the numbers of these score reports, but it is important to look beyond the numbers and the scatter plots at the bigger picture. Three facts were important in the evaluation of the UMass system, and only one of them is visible in the score reports:

```
0. MESSAGE ID                     TST1-MUC3-0004
1. TEMPLATE ID                    1
2. DATE OF INCIDENT               29 AUG 89
3. TYPE OF INCIDENT               ATTEMPTED BOMBING
4. CATEGORY OF INCIDENT           TERRORIST ACT
5. PERPETRATOR: ID OF INDIV(S)    "MAN"
                                  "WOMAN"
6. PERPETRATOR: ID OF ORG(S)      -
7. PERPETRATOR: CONFIDENCE        -
8. PHYSICAL TARGET: ID(S)         "ANTIOQUIA LIQUEUR PLANT"/
                                  "LIQUEUR PLANT"
9. PHYSICAL TARGET: TOTAL NUM     1
10. PHYSICAL TARGET: TYPE(S)      COMMERCIAL: "ANTIOQUIA LIQUEUR
                                         PLANT"/
                                              "LIQUEUR PLANT"
11. HUMAN TARGET: ID(S)           "PEOPLE"
12. HUMAN TARGET: TOTAL NUM       PLURAL
13. HUMAN TARGET: TYPE(S)         CIVILIAN: "PEOPLE"
14. TARGET: FOREIGN NATION(S)     -
15. INSTRUMENT: TYPE(S)           *
16. LOCATION OF INCIDENT          COLOMBIA: ANTIOQUIA (DEPARTMENT)
17. EFFECT ON PHYSICAL TARGET(S)  NO DAMAGE: "ANTIOQUIA LIQUEUR
                                        PLANT"/
                                              "LIQUEUR PLANT"
18. EFFECT ON HUMAN TARGET(S)     NO INJURY OR DEATH: "PEOPLE"
```

Figure 2: A Sample Answer Key

(1) The UMass/MUC-3 system was relatively successful. UMass posted the highest combined scores for recall and precision of all the systems tested.

(2) The UMass/MUC-3 system was relatively expensive to build. We estimated that 2.25 person/years of effort went into MUC-3. This represents more effort than any of the other site would admit to, and twice as much as the average effort level underlying all the MUC-3 systems.

(3) The graduate students who implemented the UMass/MUC-3 system had no desire to ever build anything like it again. Their labor was time-consuming and tedious. They established the viability of the UMass approach relative to other approaches, but with a human labor factor that threw into question the practicality of the technology.

One of my colleagues at UMass stopped me in the hallway shortly after the MUC-3 results had been released. He congratulated me on our strong showing and then asked me a question I couldn't possibly answer. He asked, "Did your group do so well because you have the best system, or because you worked harder than the other groups?"

This was a very good question and I still can't answer it. How do you quantify the effort that goes into a MUC-3 system? Do you count the time it took to develop the basic underlying approach (considering, for example, the accumulated of efforts of dozens of programmers exploring alternative sentence analysis techniques over a period of 20 years?) Do you count the time it took to construct the first two implementations that preceded the current implementation? Do you try to account for the fact that one of your programmers first learned LISP only a year ago while someone else came into the project with 10 years of LISP experience? Do you try to factor in the time it took to build an adequate computer environment with appropriate equipment and infrastructure support? The more you think about it, the more it seems that any attempt to quantify the effort behind a particular system implementation is doomed from the start.

On the other hand, there is a real world out there that responds to real economic realities and real cost constraints. The word from that world is that any system requiring as much as a year of development time is not viable in a competitive and time-sensitive marketplace. Portability from one domain to another is critical, and practical text extraction systems need to be customizable within a 6-month time frame (or less). So despite our apparent success with MUC-3, we had a serious problem on our hands. How were we supposed to construct these systems quickly? And even more challenging was the prospect of teaching others how to do what we knew how to do. A computational technology with too many variables is a little like alchemy. A few will claim that it works, but most people will shake their heads and look for other solutions.

SLOT	POS	ACT	COR	PAR	INC	ICR	IPA	SPU	MIS	NON	REC	PRE	OVG	FAL
template-id	113	215	107	0	0	0	0	108	6	17	95	50	50	
incident-date	109	103	56	21	26	0	21	0	6	4	61	64	0	
incident-type	113	107	77	20	10	0	20	0	6	0	77	81	0	0
category	81	67	55	0	8	0	0	4	18	28	68	82	6	8
indiv-perps	95	54	27	4	10	3	4	13	54	43	30	54	24	
org-perps	68	51	35	0	6	0	0	10	27	45	51	69	20	
perp-confidence	68	51	20	3	18	0	3	10	27	45	32	42	20	4
phys-target-ids	54	30	14	3	5	4	3	8	32	74	29	52	27	
phys-target-num	37	20	13	0	6	0	0	1	18	75	35	65	5	
phys-target-types	54	30	15	3	4	5	3	8	32	74	30	55	27	1
human-target-ids	144	95	50	14	17	4	14	14	63	16	40	60	15	
human-target-num	92	76	45	1	25	0	1	5	21	16	49	60	6	
human-target-types	144	95	54	21	6	2	21	14	63	16	45	68	15	1
target-nationality	18	6	4	1	0	3	1	1	13	99	25	75	17	0
instrument-types	25	11	6	0	1	0	0	4	18	84	24	54	36	0
incident-location	113	107	56	37	14	0	0	0	6	0	66	70	0	
phys-effects	36	18	12	2	2	3	2	2	20	89	36	72	11	0
human-effects	55	34	14	7	2	3	7	11	32	72	32	51	32	1
MATCHED ONLY	1361	1170	660	137	160	27	100	213	404	751	54	62	18	
MATCHED/MISSING	1419	1170	660	137	160	27	100	213	462	797	51	62	18	
ALL TEMPLATES	1419	1929	660	137	160	27	100	972	462	1926	51	38	50	
SET FILLS ONLY	594	419	257	57	51	16	57	54	229	507	48	68	13	0

Scoring Key:

POS (POSSIBLE) - the number of slot fillers according to the key target templates
ACT (ACTUAL) - the number of slot fillers generated by the system (= COR + PAR + INC + SPU)
COR (CORRECT) - the number of correct slot fillers generated by the system
PAR (PARTIAL) - the number of partially correct slot fillers generated by the system
INC (INCORRECT) - the number of incorrect slot fillers generated by the system
ICR (INTERACTIVE CORRECT) - the subset of COR judged correct during interactive scoring
IPA (INTERACTIVE PARTIAL) - the subset of PAR judged partially correct during interactive scoring
SPU (SPURIOUS) - the number of spurious slot fillers generated by the system
MIS (MISSING) - the number slot fillers erroneously not generated by the system
NON (NONCOMMITTAL) - the number of slots that were correctly left unfilled by the system
REC (RECALL) - the ratio of COR plus (.5 x) PAR slot fillers to POS slot fillers
PRE (PRECISION) - the ratio of COR plus (.5 x) PAR slot fillers to ACT slot fillers
OVG (OVERGENERATION) - the ratio of SPU slot fillers to ACT slot fillers
FAL (FALLOUT) - the ratio of INC plus SPU slot fillers to the number of possible incorrect slot fillers (a complex formula)

Figure 3: The Official UMass Score Report from MUC-3

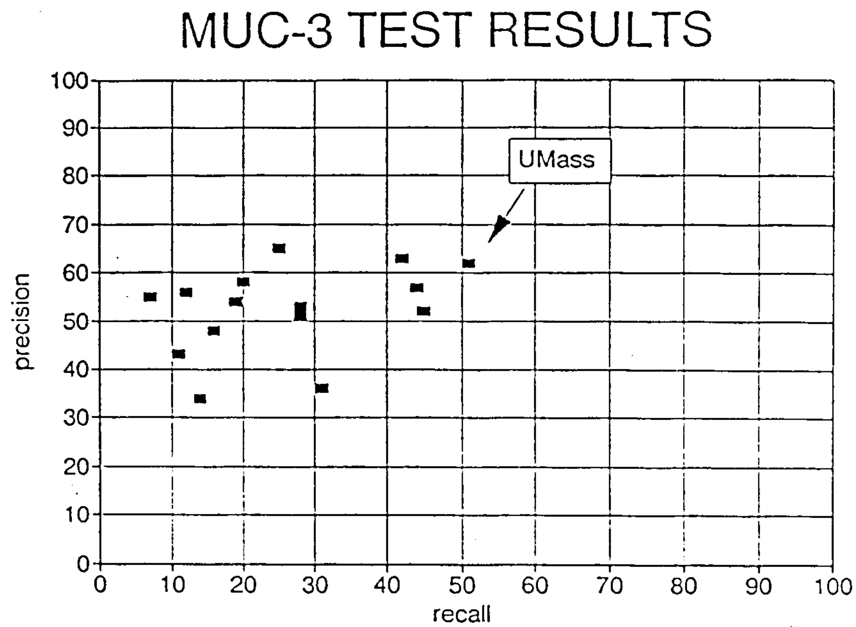

Figure 4: Overall Recall/Precision Results for all the MUC-3 sites

Setting aside the questions of portability and scalability, UMass/MUC-3 was nevertheless notable as the only MUC system that had managed to achieve high performance levels while avoiding involved syntactic sentence analysis. Back in the days of The Great Debate, Schank argued that the linguists' preoccupation with syntactic parse trees was rather besides the point, and that the considerable intellectual effort associated with syntactic parsing was misguided. Schank never did manage to turn the field of computational linguistics around, but MUC-3 finally gave us some hard evidence in support of Schank's early warnings.[12]

MUC-4: Some More Lessons (1992)

As if scaling up weren't hard enough, we were now expected to do it quickly and effortlessly. Mercifully, MUC-4 was organized around the familiar domain of terrorism, and this afforded us a little time to ponder the problems of scaling-up. We were determined to try to work smarter for MUC-4, if only because we didn't have the labor pool available to us for MUC-4 that we had for MUC-3. The post doc who had worked on MUC-3 was offered a professorship at Pace University and left UMass. I had promised the graduate students who worked on MUC-3 that I would never ask them to do anything like MUC-3 again. The only other MUC-3 survivor was our undergraduate assistant, but it's not sporting to lean too hard on an undergraduate. I was supporting a new first year graduate student, but he was buried by course requirements and therefore out of bounds for any sort of sustained development effort. And I was working with one other advanced graduate student, but he had just joined the project, so he had no experience with MUC-3. I didn't see how it we could move into MUC-4 with any sort of a development effort like the one we mounted for MUC-3.

---

[12]The status of syntactic parse trees was challenged even more dramatically the following year at MUC-4 where a second upstart system performed very well without any recourse to traditional syntactic parsing [Sundheim 1992].

CIRCUS was the sentence analyzer at the heart of the UMass/MUC-3 system. Although the basic operations of CIRCUS are reasonably simple, the dictionary that CIRCUS uses is not. CIRCUS dictionary definitions are somewhat more complicated than the definitions found in your standard collegiate dictionary, and we had estimated that 9 person/months went into the construction of the dictionary that we used for MUC-3. This dictionary was what made the UMass/MUC-3 work or not work on any given sentence. But the process of constructing such a dictionary could only be called a black art. It wasn't even clear that this art could be handed down from one graduate student to another after a lengthy apprenticeship.

In the midst of all this difficulty, something completely unexpected occurred. Ellen Riloff (one of the advanced graduate students who survived MUC-3) reminded me about something she had worked on briefly the summer after MUC-3. She had designed and implemented an experimental system that plodded its way through the entire development corpus attempting to construct dictionary definitions based on example sentences encountered in the corpus. She knew that her construction tool was coming up with a lot of bogus definitions, and it was not at all clear how to improve its hit rate. Ellen was on the verge of throwing the whole thing away because it produced so many bad definitions. But I was curious to see what we might salvage if we took all the definitions it produced and eliminated the bogus ones by visual inspection. So I cornered my first-year graduate student and asked him to sift through the 1,356 definitions proposed by Ellen's construction tool. His job was to sort them into two groups: the good ones and the bad ones.

Four weeks before the final evaluation for MUC-4, the manual sifting task was completed and we ran a test run using the (filtered) dictionary definitions. Much to our surprise, this semi-automated dictionary reproduced 95% of the functionality of our hand-crafted MUC-3 dictionary. When asked how long it had taken to complete the manual filtering task, we found out that it had taken less than 8 hours to sift through all the definitions.[13] Our hand-crafted dictionary from MUC-3 was the result of about 1500 hours of highly skilled and experienced labor on the part of three individuals. In the space of 8 hours, an inexperienced first-year graduate student had come very close to duplicating the functionality of the UMass/MUC-3 dictionary. We quickly honored the responsible code with a descriptive moniker, and dubbed Ellen's dictionary construction tool AutoSlog.[14]

So we showed up at MUC-4, site report in hand [Lehnert et al. 1992b], feeling very pleased that we could respond to questions about the portability and scalability of our approach that had been raised the year before. Not only did our approach yield strong performance, but it might actually be possible for even unscruffy people to build systems based on our scruffy technology within a reasonable time frame. In the space of 8 hours, our admittedly ad hoc UMass/MUC-3 dictionary was transformed into something that could be scaled up and ported out. This was a whole new ballgame.

Rapid Cross-Fertilization with MUC (June 1992)

As an experienced academic, I have come to understand that people who have worked on the intricate complexities of reversible bleemers for 15 or 20 years are very likely to keep working on reversible bleemers as long as they can (barring the usual mid-life crisis or total career flip). No amount of heated argumentation, well-constructed debates, or diplomatic reasoning is likely to make a dent in the

---

[13] Stephen Soderland, the first-year student who did the sifting work, admitted sometime later that he was able to move so quickly through the definitions because he knew that "whatever it was he was doing, it couldn't be very important."

[14] The name bestowed upon a piece of software is every bit as important as the names we give our children and pets. We live with these names for years on end, fold them into countless publications, and introduce them many times over in public talks and private seminars. The really successful ones make it into the textbooks. A good name should be memorable, suggestive in appropriate ways, and easy to spell.

professional beliefs or foundational assumptions of a tenured professor. Academic freedom should never be confused with intellectual flexibility. I have seen energetic researchers drop an entire field in order to pick up on a new one, but I don't think I have ever seen anyone turn 180 degrees within the confines of a fixed community. To do so would call into question one's hard-won professional credibility, intellectual stamina, and political street smarts.

For all of these reasons, academic conferences are often predictable gatherings. There may be new results, to be sure, but there are almost no surprises. A lot of rhetoric is exchanged, but few minds are ever changed. After a few years on the conference circuit, it can all begin to feel like sleep walking. Having acquired the necessary sleep-walking skills required of my profession, I had become fairly cynical about a lot of conferences, but especially cynical about conferences devoted to computational linguistics or natural language processing.[15]

So imagine my surprise to stumble upon a gathering of natural language processing researchers where researchers who had never showed an interest in one another's work are suddenly talking to each other and asking one another substantive questions. Imagine the excitement of a meeting where the pursuit of ideas overcomes the mammalian instinct to preserve predictable social orders! What I had seen for the first time at MUC-3 was happening once again at MUC-4. The shared goal of an ambitious working system was steadily eroding otherwise impenetrable ego boundaries.

To see this, consider the MUC-4 site report of a highly respected commercial laboratory. A survivor of MUC-3, they explained that it was their own profound dissatisfaction with their MUC-3 performance that had led them to experiment with a completely new and radically different MUC-4 system. Moreover, in so doing, they had succeeded in producing a system that showed tremendous promise. Here is an excerpt from their MUC-4 site report:

> " ... The inspiration for FASTUS was threefold. First, we were struck by the strong performance that the group at the University of Massachusetts got out of a fairly simple system [Lehnert et al. 1991]. It was clear they were not doing anything like the depth of preprocessing, syntactic analysis, or pragmatics that was being done by the systems at SRI, General Electric, or New York University. They were not doing a lot of processing. They were doing the *right* processing."
> [p. 268, Sundheim 92]

The authors of this report had discovered the 80-20 rule. With a heightened sensitivity for the difference between essential and inessential capabilities, they engineered a new way to get at the essentials. In so doing, they left behind a number of sophisticated components including a syntactic chart parser, a statistical relevance filter (and keyword antifilter), and an abductive inference module. It was not surprising to see another mechanism duplicate the UMass performance levels.[16] What was impressive was the extremely fast development cycle that this site achieved with their new system. Having cut to the heart of what mattered, they were able to implement a MUC-4 system in the space of one month. Their overall performance was neck and neck with UMass at the time of the final evaluation, and the shape of their development curve showed no signs of leveling out as they moved toward the final evaluation. Everyone at the MUC-4 meeting wished they could see what another week or month would have yielded for this adventurous research team.

---

15 It would be unfair to suggest that all academic gatherings are quite this dismal. In fact, this grim picture does not apply at all to interdisciplinary meetings where researchers with different backgrounds and different research methodologies meet to compare notes and search for common ground. My impatience with conferences is really directed at those conferences which are narrowly defined and therefore likely to attract the same group of people year after year after year. These gatherings take on the surrealistic feel of a 25-year TV reunion with the cast of the Partridge Family.

16 Indeed, a number of systems were operating at levels comparable to UMass at MUC-4, and one site pulled ahead of us.

It was genuinely exciting to see such strong success achieved by a completely new system. It was a clear demonstration that the MUC meetings were working as a device for intellectual cross fertilization. Participating MUC sites were free to shop around for good ideas, incorporating whatever goodies they can, or rebuilding an entire system if need be. By building on one another's successes we seemed to be converging on an optimal technology much more readily than we would if we were each left alone to work in isolation. It is exciting to be a part of that competitive/cooperative community process. More importantly, the pursuit of a practical working system can also work as a highly effective stimulus for basic research as well.[17]

But Does it Really Understand?

Sooner or later somebody always has to ask this. There are many ways to answer. At one level, we can ask if the system succeeds at the task it was designed to perform. In our case, we wanted to extract relevant information about terrorist incidents. To see how we're doing on that score, let's take a look at a sample text and the resulting system output generated by the UMass/MUC-4 system.[18] The following text was discussed at the MUC-4 meeting by each of the participating sites during the system walk through presentations:

```
TST2-MUC4-0048
    SAN SALVADOR, 19 APR 89 (ACAN-EFE) -- [TEXT] SALVADORAN PRESIDENT-ELECT
ALFREDO CRISTIANI CONDEMNED THE TERRORIST KILLING OF ATTORNEY GENERAL ROBERTO
GARCIA ALVARADO AND ACCUSED THE FARABUNDO MARTI NATIONAL LIBERATION FRONT
(FMLN) OF THE CRIME. LEGISLATIVE ASSEMBLY PRESIDENT RICARDO VALDIVIESO AND
VICE PRESIDENT-ELECT FRANCISCO MERINO ALSO DECLARED THAT THE DEATH OF THE
ATTORNEY GENERAL WAS CAUSED BY WHAT VALDIVIESO TERMED THE GUERRILLAS'
"IRRATIONAL VIOLENCE." GARCIA ALVARADO, 56, WAS KILLED WHEN A BOMB PLACED BY
URBAN GUERRILLAS ON HIS VEHICLE EXPLODED AS IT CAME TO A HALT AT AN
INTERSECTION IN DOWNTOWN SAN SALVADOR.
...
"WE HAVE TO CONDEMN THIS INCIDENT, IT IS A GUERRILLA ACT," ALFREDO CRISTIANI,
NATIONALIST REPUBLICAN ALLIANCE (ARENA) PRESIDENT-ELECT, WHO WILL REPLACE
CHRISTIAN DEMOCRAT JOSE NAPOLEON DUARTE ON 1 JUNE, STATED. CRISTIANI SAID THAT
"THESE ARE THE RISKS FACED BY SOMEONE WHO ENFORCES THE LAW."  HE NOTED THAT
"THE GUERRILLAS' IRRATIONAL ATTITUDE MAKES IT INCREASINGLY DIFFICULT TO
BELIEVE THEY WANT PEACE." ACCORDING TO CRISTIANI, THE ATTACK TOOK PLACE
BECAUSE ATTORNEY GENERAL GARCIA ALVARADO WARNED THAT "HE WOULD TAKE MEASURES
AGAINST URBAN TERRORISTS."

VICE PRESIDENT-ELECT FRANCISCO MERINO SAID THAT WHEN THE ATTORNEY GENERAL'S
CAR STOPPED AT A LIGHT ON A STREET IN DOWNTOWN SAN SALVADOR, AN INDIVIDUAL
PLACED A BOMB ON THE ROOF OF THE ARMORED VEHICLE. "THE DRIVER TOLD THE
ATTORNEY GENERAL ABOUT THE BOMB.  THE VEHICLE SWERVED AND THE BOMB EXPLODED,
CAUSING THE TOP OF THE VEHICLE TO COLLAPSE ON THE ATTORNEY GENERAL'S HEAD,"
MERINO STATED.
```

---

[17] Claire Cardie (the other graduate student who had worked on MUC-3 along with Ellen Riloff) produced an impressive collection of research papers in the year that followed MUC-3 [Cardie 1992a, 1992b, 1992c and 1992d]. Ellen Riloff is producing equally impressive work in the year following MUC-4 [Riloff and Lehnert 1992], [Riloff and Lehnert 1993], [Riloff 1993a] [Riloff 1993b].

[18] MUC-3 output and MUC-4 output were subject to slightly different formatting guidelines, so the output templates shown here do not completely correspond with the answer key format presented earlier in Figure 2.

GUERRILLAS ATTACKED MERINO'S HOME IN SAN SALVADOR 5 DAYS AGO WITH EXPLOSIVES. THERE WERE SEVEN CHILDREN, INCLUDING FOUR OF THE VICE PRESIDENT'S CHILDREN, IN THE HOME AT THE TIME. A 15-YEAR-OLD NIECE OF MERINO'S WAS INJURED.

"THESE INCIDENTS," CRISTIANI SAID, "FRANKLY CAUSE US TO BECOME MORE AWARE OF THE FACT THAT WE MUST NOT PERMIT TERRORIST ACTIONS TO OCCUR IN EL SALVADOR." THE PRESIDENT-ELECT RULED OUT THE POSSIBILITY THAT THESE ATTACKS "WILL PREVENT THE INAUGURAL CEREMONY FROM TAKING PLACE."

"I AM CERTAIN THAT THE INAUGURATION WILL BE ON 1 JUNE. WE WILL NOT JUMP OVERBOARD OR MAKE A RUN FOR IT. WE KNOW WHAT WE ARE UP AGAINST AND WILL GO ON," HE STATED.

CRISTIANI SAID THE GUERRILLA ATTACKS ARE INTENDED TO PROMPT A GOVERNMENT AND MILITARY REACTION SO THE FMLN CAN "EXPLOIT IT" ABROAD TO "POLITICALLY ISOLATE THE NEW GOVERNMENT."

RICARDO VALDIVIESO, PRESIDENT OF THE LEGISLATIVE ASSEMBLY AND AN ARENA LEADER, SAID THE FMLN AND ITS "FRONT" GROUPS ARE RESPONSIBLE FOR THE "IRRATIONAL VIOLENCE THAT KILLED ATTORNEY GENERAL GARCIA." VALDIVIESO SAID THE LEGISLATIVE ASSEMBLY WILL APPROVE DRASTIC LAWS TO "HALT THE WAVE OF VIOLENCE." HE SAID THE ATTORNEY GENERAL "WAS APOLITICAL, WORKED FOR JUSTICE, AND DID NOT DESERVE TO DIE LIKE THAT."

ACCORDING TO THE POLICE AND GARCIA ALVARADO'S DRIVER, WHO ESCAPED UNSCATHED, THE ATTORNEY GENERAL WAS TRAVELING WITH TWO BODYGUARDS. ONE OF THEM WAS INJURED. THE ATTORNEY GENERAL'S BODY WAS DESTROYED BY THE BOMB THAT EXPLODED OVER HIS HEAD.

NO GROUP HAS CLAIMED CREDIT FOR THE ATTACK YET, BUT POLICE SOURCES CLAIM IT "IS CHARACTERISTIC OF THE FMLN URBAN COMMANDOS." THE SAME SOURCES CONFIRMED THAT GARCIA ALVARADO HAD BEEN THREATENED ON SEVERAL OCCASIONS BY SALVADORAN URBAN GUERRILLAS.

MOMENTS AFTER THE ATTACK, ARMY AND POLICE UNITS CORDONED OFF THE AREA AND BEGAN AN ALL-OUT MILITARY OPERATION TO FIND THOSE RESPONSIBLE.

GARCIA ALVARADO, FATHER OF SIX, WAS APPOINTED ATTORNEY GENERAL ON 23 DECEMBER 1988. HE WAS CONSIDERED TO BE CLOSELY LINKED TO ARENA. ON SEVERAL OCCASIONS, HOWEVER, HE SAID HE DID NOT REPRESENT ANY PARTY AND WAS CARRYING OUT HIS JOB "IMPARTIALLY AND WITH THE INTENTION OF ENFORCING THE COUNTRY'S LAWS."

There are two answer keys associated with this text. One describes the bombing of the armored vehicle and one describes the bombing of Merino’s home. UMass/MUC-4 generated templates for each of these incidents. The first template is shown in Figure 5.

0. MESSAGE: ID	TST2-MUC4-0048	
1. MESSAGE: TEMPLATE	1	;correct
2. INCIDENT: DATE	- 19 APR 89	;correct
3. INCIDENT: LOCATION	EL SALVADOR	;partial
4. INCIDENT: TYPE	BOMBING	;correct
5. INCIDENT: STAGE OF EXEC.	ACCOMPLISHED	;correct
6. INCIDENT: INSTRUMENT ID	"BOMB"	;correct
7. INCIDENT: INSTRUMENT TYPE	BOMB: "BOMB"	;correct
8. PERP: INCIDENT CATEGORY	TERRORIST ACT	;correct
9. PERP: INDIVIDUAL ID	"URBAN GUERRILLAS"	;correct
10: PERP: ORGANIZATION ID	"FARABUNDO MARTI NATIONAL LIBERATION FRONT"	;correct
11: PERP: ORG CONFIDENCE	SUSPECTED OR ACCUSED BY AUTHORITIES: "FARABUNDO MARTI NATIONAL LIBERATION FRONT"	;correct
12: PHYS TGT: ID	"ARMORED VEHICLE"	;correct
13: PHYS TGT: TYPE	TRANSPORT VEHICLE: "ARMORED VEHICLE"	;correct
14: PHYS TGT: NUMBER	1: "ARMORED VEHICLE"	;correct
15: PHYS TGT: FOREIGN NATION	-	;N/A
16. PHYS TGT: EFFECT	DESTROYED: "ARMORED VEHICLE"	;partial
17: PHYS TGT: TOTAL NUMBER	-	;N/A
18: HUM TGT: NAME	"ROBERTO GARCIA ALVARADO"	;correct
19: HUM TGT: DESCRIPTION	"ATTORNEY GENERAL": "ROBERTO GARCIA ALVARADO"	;correct/missing
20: HUM TGT: TYPE	GOVERNMENT OFFICIAL: "ROBERTO GARCIA ALVARADO"	;correct/missing
21: HUM TGT: NUMBER	1: "ROBERTO GARCIA ALVARADO"	;correct/missing
22. HUM TGT: FOREIGN NATION	-	;N/A
23: HUM TGT: EFFECT	DEATH: "ROBERTO GARCIA ALVARADO"	;correct/missing
24: HUM TGT: TOTAL NUMBER	-	;N/A

Figure 5: The First Output Template for TST2-MUC4-0048

We did fairly well on this template. We missed San Salvador as the location within El Salvador, we said the vehicle was destroyed instead of damaged, and we missed 3 human targets (the driver who was not hurt, and the 2 bodyguards, one of whom was injured). All the other slots were correctly filled.

The summary portion of the score report for this single template is shown in Figure 6.

	POS	ACT	COR	PAR	INC	ACR	IPA	SPU	MIS	NON	REC	PRE	OVG
inc-total	6	6	5	1	0	0	0	0	0	0	92	92	0
perp-total	4	4	4	0	0	0	0	0	0	0	100	100	0
phys-tgt-total	4	4	3	1	0	0	0	0	0	2	88	88	0
hum-tgt-total	14	5	5	0	0	0	0	0	9	2	36	100	0
TOTAL	28	19	17	2	0	0	0	0	9	4	64	95	0

Figure 6: A Score Report for the Vehicle Bombing Template

The second output template generated by UMass/MUC-4 is shown in Figure 7.

0. MESSAGE: ID	TST2-MUC4-0048	;correct
1. MESSAGE: TEMPLATE	3	;correct
2. INCIDENT: DATE	14 APR 89	;correct
3. INCIDENT: LOCATION	EL SALVADOR: SAN SALVADOR	;correct
4. INCIDENT: TYPE	BOMBING	;correct
5. INCIDENT: STAGE OF EXEC.	ACCOMPLISHED	;correct
6. INCIDENT: INSTRUMENT ID	"EXPLOSIVES"	;correct
7. INCIDENT: INSTRUMENT TYPE	BOMB: "EXPLOSIVES"	;correct
8. PERP: INCIDENT CATEGORY	TERRORIST ACT	;correct
9. PERP: INDIVIDUAL ID	"GUERRILLAS"	;correct
10: PERP: ORGANIZATION ID	-	;missing
11: PERP: ORGANIZATION CONF.	-	;missing
12: PHYS TGT: ID	"MERINO'S HOME"	;correct
13: PHYS TGT: TYPE	CIVILIAN RESIDENCE: "MERINOS HOME"	;incorrect
14: PHYS TGT: NUMBER	1: "MERINO'S HOME"	;correct
15: PHYS TGT: FOREIGN NATION	-	;N/A
16. PHYS TGT: EFFECT	-	;N/A
17: PHYS TGT: TOTAL NUMBER	-	;N/A
18: HUM TGT: NAME	-	;N/A
19: HUM TGT: DESCRIPTION	"15-YEAR-OLD-NIECE"	;correct/missing
20: HUM TGT: TYPE	CIVILIAN: "15-YEAR-OLD-NIECE"	;correct/missing
21: HUM TGT: NUMBER	1: "15-YEAR-OLD-NIECE"	;correct/missing
22. HUM TGT: FOREIGN NATION	-	;N/A
23: HUM TGT: EFFECT	INJURY: "15-YEAR-OLD-NIECE"	;correct
24: HUM TGT: TOTAL NUMBER	-	;missing

Figure 7: The Second Output Template for TST2-MUC4-0048

Here we fail in three places. We have no perpetrator organization, we miss the physical target type for Merino's home (it should have been GOVERNMENT OFFICE OR RESIDENCE), and we are missing the 7 children that were human targets (this is one of the few texts where a TOTAL NUMBER slot should receive a value).

The summary portion of the score report for this single template is shown in Figure 8.

	POS	ACT	COR	PAR	INC	ACR	IPA	SPU	MIS	NON	REC	PRE	OVG
inc-total	6	6	6	0	0	0	0	0	0	0	100	100	0
perp-total	4	2	2	0	0	0	0	0	2	0	50	100	0
phys-tgt-total	3	3	2	0	1	0	0	0	0	3	67	67	0
hum-tgt-total	11	4	4	0	0	0	0	0	7	2	36	100	0
TOTAL	24	15	14	0	1	0	0	0	9	5	58	93	0

Figure 8: A Score Report for the Home Bombing Template

Because we generated a third (spurious) template for a threat that is mentioned near the end of the story, our overall score report for TST2-MUC4-0048 does not reflect the very strong precision present in these first two templates. Figure 9 shows the final score report for TST2-MUC4-0048 when all three templates are averaged together.

This shows how much negative impact spurious templates have on precision if a system is generating one spurious template for every two good templates. If we had generated a summary score report based on only two templates instead of three, our All Templates precision would have been 94 rather than 76.

Overall, TST2-MUC4-0048 illustrated the UMass/MUC-4 system when it is working fairly well and not making major errors. Most of our recall loss resulted from a failure to recognize relevant information in sentence 12 (the 7 children), and sentences 21-22 (the driver and 2 bodyguards).

	POS	ACT	COR	PAR	INC	ACR	IPA	SPU	MIS	NON	REC	PRE	OV
inc-total	12	16	11	1	0	0	0	4	0	2	96	72	25
perp-total	8	7	6	0	0	0	0	1	2	3	75	86	14
phys-tgt-total	7	7	5	1	1	0	0	0	0	11	78	78	0
hum-tgt-total	25	12	9	0	0	0	0	3	16	8	36	75	25
MATCHED/MISSING	52	34	31	2	1	0	0	0	18	9	62	94	0
MATCHED/SPURIOUS	52	42	31	2	1	0	0	8	18	24	62	76	19
MATCHED ONLY	52	34	31	2	1	0	0	0	18	9	62	94	0
ALL TEMPLATES	52	42	31	2	1	0	0	8	18	24	62	76	19
SET FILLS ONLY	23	16	14	1	1	0	0	0	7	5	63	91	0
STRING FILLS ONLY	15	10	10	0	0	0	0	0	5	1	67	100	0

P&R	2P&R	P&2R
68.29	72.72	64.37

Figure 9: The Message Score Report for TST2-MUC4-0048

According to the answer keys, we've missed about a third of the content that we should have gotten. However, one could reasonably argue that not all human targets are equal and Attorney Generals are normally more important than bodyguards. If that's the case, our recall for this story is better than the recall figures suggest. As far as precision goes, we are showing exceptionally strong precision for the two templates described here. So let's be generous for a moment, and assume this is the state-of-the-art in text extraction systems. Then we seem to be getting a lot of what we set out to get. But does it really understand?

I'd say that the UMass/MUC-4 system understands terrorism about as well as SAM understood restaurants. It is limited by its dictionary, its knowledge, and its ability to handle complicated sentences. When the UMass/MUC-4 system saw ".. THE KILLING OF TWO BIRDS ..." during the MUC-4 test runs, the system fell hook, line, and sinker. With no "bird" in our lexicon, we assumed we had a proper name, as in "... THE KILLING OF TWO JESUITS ...." and so a murder was dutifully recorded in our database of terrorist incidents. That is clearly not the sort of a mistake a person would make.

On the other hand, UMass/MUC-4 runs rings around SAM when it comes to robustness and generality. We can unravel sentences of far greater complexity than SAM ever could, and we generally manage to make sense of the things we need in spite of the fact that our dictionary is sparse. Would we say that our sentence analysis is consistent with human language processing capabilities?

This is hard to answer, but I am inclined to say yes with some caveats. We are certainly interested in the cognitive validity of our sentence analyzer [Cardie and Lehnert 1991]. To ask the question in the context of the MUC task, one must somehow take into account the implications of an incomplete dictionary and the severe attention deficit disorder that limits the system's comprehension to incidents of a terrorist nature. This makes comparisons to humans difficult, although perhaps not so strained if you consider the language processing capabilities of small children. A very young child is exposed to constant streams of language that do not, for the most part, make sense. Still, an occasional phrase is picked up without difficulty (... *eat some ice cream* ...) and a selectional mechanism is probably highly effective in filtering out anything that is not motivated by self interest. The UMass/MUC-4 system is a lot like a very small child who is oddly preoccupied with terrorism. Given this peculiar orientation, the UMass/MUC-4 system is a highly robust and largely effective language processing system.

God is in the Details (and lots of them ...)

Given the limitations of our ad hoc prototypes in the 70s, how did we manage to get from SAM to MUC? Granted, MUC assumes a limited domain with strictly focused extraction goals, but the MUC evaluations were nevertheless based on previously unseen texts. Any success on blind test sets must signify some success with robust language processing. How did we progress from our brittle prototypes to robust systems? Did we unearth some general principles of natural language that eluded us 20 years ago? Did we embrace some new approach to sentence analysis that is fundamentally different from the one we worked with earlier? How did we get beyond the reverse engineering that characterized SAM in 1975?

Strange to say, I must admit that we haven't gotten beyond the reverse-engineering problem at all. But I do think that we understand something today that we didn't understand in 1975. Namely, the answer to the reverse-engineering problem is *bigger* and *better* reverse-engineering. In 1975 we practiced reverse-engineering in-the-small. In 1992 we are just beginning to discover reverse-engineering in-the-large. If this is a surprising development, it is surprising for the same reasons that AutoSlog's success was surprising. AutoSlog is a tour de force in reverse-engineering. When AutoSlog works its way through the corpus looking for possible dictionary definitions, it tries to make useful generalizations on the basis of single sentences. Some are clearly reasonable while others are much less so. The ability to move from a lot of single examples to something resembling general competence is the essence of reverse-engineering in-the-large. Here are some examples of text from the development corpus along with the conclusions drawn by AutoSlog:

```
... A GROUP OF ARMED INDIVIDUALS WEARING SKI MASKS ROBBED A BUSINESSMAN
     ...
==> the direct object of the verb "to rob" (active voice) is the victim
     of a robbery
... THE ASSISTANT SECRETARY OF THE NATIONAL REVOLUTIONARY MOVEMENT
     DISAPPEARED ON JAN 12 ...
==> the subject of the verb "to disappear" (active voice) is the victim
     of a kidnapping

... SALVADORAN PRESIDENT ALFREDO CRISTIANI IMPLICATED 4 OFFICERS ...
==> the direct object of the verb "to implicate" (active voice) is the
     perpetrator of a terrorist act

... LAST NIGHT THERE WERE FEWER ATTACKS ON STORES ...
==> the object of "on" following the noun "attacks" is the target of an
     attack
... THE GROUP WAS TRAVELING IN A 4-WHEEL DRIVE VEHICLE ...
==> the object of "in" after the verb "to travel" (passive voice [sic])
     is the target of an attack
...  REPORTED THAT DYNAMITE STICKS WERE HURLED FROM A CAR ...
==> the subject of the verb "to hurl" (passive voice) is the instrument
     of an attack
... A BOMB WAS PLACED OUTSIDE GOVERNMENT HOUSE IN THE PARKING LOT....
==>   the subject of the verb "to place" (passive voice) is the
     instrument of a bombing
```

These are examples of AutoSlog definitions we would want to keep. The bad ones tend to guess that any subject of the verb "to be" is a terrorist perpetrator, along with numerous other bad assumptions.[19]

Looking at these definitions, it is clear that they may work for this particular task in this particular domain, but there is little here of general value. If we proposed 10 of these definitions for the sake of handling 10 sample sentences we would be back in the ad hoc prototype game. But something shifts when we move from 10 sample sentences to 100 previously unseen texts. The AutoSlog dictionary that handles blind test sets contains the same type of ad hoc definitions. There are just a lot more of them. Most interestingly, if this dictionary can be derived automatically (or semi-automatically), then the ad hoc criticism falls away. But in fact, we're still reverse-engineering the dictionary: we're just doing it in a way that makes the technology portable and scalable.

As one warms up to the world according to AutoSlog, two questions come to mind.

Q1: Are the possibilities endless? If so, we're in trouble.

Q2: Some of these patterns might be valid but only for this domain and only for this task orientation. As soon as you change domains and/or tasks, you'll have to start all over again.

We can answer Q1 because we tracked the number of new definitions that AutoSlog generated as it moved through the development corpus. The number of novel definitions dropped off as we got further and further into the corpus. By the time we had gotten through 1200 texts, AutoSlog was finding only about one third as many new definitions that it found at the beginning of the corpus, suggesting that two thirds of the potential definitions lurking near the end of the corpus had already been uncovered earlier in the corpus.

Q2 strikes at the heart of all practical natural language processing systems. No matter what approach is taken, no natural language system can claim to tackle general language in an open-ended domain or task orientation. Natural language processing applications are effective only when the application circumscribes a finite domain that can be covered by a limited lexicon and limited domain knowledge. When we talk about scaling and porting, we are still assuming the context of a "reasonable" task orientation where workable limits are in place.

Although it might sound risky to assume that the verb "to place" signals a description of a bombing, that turns out to be a highly reliable linguistic cue within the world of terrorism texts[20]. Similarly, we never hear about the vehicle that someone is traveling in unless there was an attack on that vehicle. Each of these linguistic/conceptual signatures goes into our dictionary where they are then used to extract relevant information when we analyze a new text. In our official test run for MUC-4, 389 conceptual definitions were used to attain 47% recall on 100 test texts. Of these 389 definitions, only 20% were needed to generate 74% of that recall (a clear example of the 80-20 rule at work). So we see that it doesn't take a massive amount of knowledge to perform reasonably effective and highly robust text extraction. But it does take the right knowledge. It would be very difficult to guess at an effective set of conceptual dictionary definitions using intuition alone.

So how do we go from terrorism to the world at large? And do we have to make that leap before we can lay serious claims to psychological validity? When a system can hold its own in casual conversation at a social gathering, we will have crossed out of microworlds. When we can shift from strongly goal-oriented systems to systems that read fiction, we will have crossed that same line. Are we any closer to these goals?

---

[19]Of the 1,356 concept node definitions originally proposed by AutoSlog, only 375 were judged to be acceptable after manual inspection.

[20]The only thing that ever gets placed anywhere in the MUC-4 corpus is a bomb.

The UMass/MUC-4 system is certainly closer than SAM was. Even AutoSlog is closer to a realistic language acquisition device than any manual design effort. But AutoSlog works because it is fixated on specific kinds of information. Moreover, it uses a large development corpus in order to sensitize a dictionary to the very specific linguistic contexts in which that narrow range of information appears.

Will there be a Son-of-AutoSlog that can operate in a less goal oriented fashion? Or is language acquisition always a goal-oriented process? Is there a neutral training corpus that can serve as the basis for automated dictionary acquisition without domain bias? Children seem to acquire language in a relatively neutral fashion. But do they acquire linguistic knowledge in the absence of goals? These are fundamental questions about the nature of language and language acquisition. I don't think we have a lot of the answers.

The current state-of-the-art in information extraction systems does not have a strong analogical human counterpart. The closest analogy we have is perhaps the notion of a pathological toddler with very narrow and thoroughly adult interests. Maybe the technology underlying today's systems can only give rise to more sophisticated systems that mimic pathological children with very limited interests.

There is an intriguing connection between scalability and language acquisition. If AutoSlog is a step in the right direction, then language acquisition has nothing to do with the acquisition of complex grammars, and everything to do with the ability to comprehend, organize, and recall an extremely large store of lexical/syntactic/semantic patterns.

There are linguistic phenomena that support a view of a language acquisition as a process that is driven by highly specific examples with only the most conservative inclination to generalize. What else can explain the frozen passives of certain metaphors? We say either "they buried the hatchet" or "the hatchet was buried," but we can only say "he kicked the bucket." It never occurs to anyone (outside of linguistics departments) to say "the bucket was kicked." What prohibits us from framing this particular metaphor in the passive voice if language is truly generative?

As we continue to confront the issues of scalability and portability in our computer programs, we may be better able to answer these questions about language acquisition in both machines and humans. In the worst case, we won't come up with any definitive answers, but we will create some useful computer systems to help us cope with the Information Age. When Alan Perlis nailed the quintessential computer experience in his famous epigram, I like to think that he was describing the quest for Artificial Intelligence. Anyone who knows anything about computer programming can tell you that nothing is easy. Only the True Believers understand that everything is possible.

Bibliography

Abelson, R. P. (1981). "Constraint, Constructual, and Cognitive Science" in the *Proceedings of the Third Annual Conference of the Cognitive Science Society.* pp. 1-9.

Cardie, C. (1992a). "Learning to Disambiguate Relative Pronouns" *Proceedings, Tenth National Conference on Artificial Intelligence*, San Jose, CA. pp. 3843.

Cardie, C. (1992b). "Corpus-Based Acquisition of Relative Pronoun Disambiguation Heuristics" *Proceedings of the Thirtieth Annual Meeting of the Association for Computational Linguistics*. Newark, Delaware. pp. 216-223.

Cardie, C. (1992c). "Using Cognitive Biases to Guide Feature Set Selection" in the *Proceedings of the AAAI-92 Workshop on Constraining Learning with Prior Knowledge*. San Jose, CA. pp. 11-18.

Cardie, C. (1992d). "Using Cognitive Biases to Guide Feature Set Selection" *Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society*. pp. 743-48.

Cardie, C.T. and Lehnert, W.G. (1991). "A Cognitively Plausible Approach to Understanding Complex Syntax," *Proceedings of the Ninth National Conference on Artificial Intelligence.* Anaheim, CA. pp. 117-124.

Chomsky, N. (1965). *Aspects of the Theory of Syntax*. MIT Press.

Cullingford, R. (1978). "Script Application: Computer Understanding of Newspaper Stories", Yale University, Department of Computer Science Technical Report No. 116.

Feigenbaum, E. and J. Feldman (1963). *Computers and Thought*. McGraw Hill. New York.

Kintsch, W., J. Miller, and P. Polson. (1981). Problems of methodology in cognitive science. Paper for the Cognitive Science Symposium. Boulder, Col., July 1981.

Lehnert, W. (1975). "What Makes SAM Run? Script-Based Techniques for Question Answering," in *Proceedings of Theoretical Issues in Natural Language Processing*,, Cambridge, Massachusetts.

Lehnert, W.G., R. Williams, C. Cardie, E. Riloff and D. Fisher. (1991a). "University of Massachusetts: MUC-3 Test Results and Analysis," in *The Proceedings of the Third Message Understanding Conference*. pp. 116-119.

Lehnert, W.G., R. Williams, C. Cardie, E. Riloff and D. Fisher. (1991b). "University of Massachusetts: Description of the CIRCUS System as Used for MUC-3," in *The Proceedings of* the *Third Message Understanding Conference*e. pp. 223-233.

Lehnert, W.G. and B. Sundheim. (1991). "A Performance Evaluation of Text Analysis Technologies", *AI Magazine,* Fall 1991. pp. 81-94.

Lehnert, W.G., C. Cardie, D. Fisher, J. McCarthy, E. Riloff and S. Soderland. (1992a) "University of Massachusetts: Description of the CIRCUS System as Used for MUC-4", in *The Proceedings of the Fourth Message Understanding Conference*. pp. 282-288.

Lehnert, W.G., C. Cardie, D. Fisher, J. McCarthy, E. Riloff, and S. Soderland. (1992b). "University of Massachusetts: MUC-4 Test Results and Analysis," in *The Proceedings of the Fourth Message Understanding Conference*. pp. 151-158.

Minsky, M. (1991). "Logical versus Analogical or Symbolic versus Connectionist or Neat versus Scruffy". *AI Magazine*. Vol. 12, No. 2, Summer 1991. pp. 34-51.

Riloff, E. (1993a). "Using Cases to Represent Context for Text Classification", to appear in *Working Notes of the AAAI Spring Symposium on Case-Based Reasoning and Information Retrieval.*

Riloff, E. (1993b). "Automatically Constructing a Dictionary for Information Extraction Tasks". To appear in *Proceedings of the Eleventh National Conference for Artificial Intelligence.*

Riloff, E. and W. Lehnert. (1993). "Automated Dictionary Construction for Text Extraction". In *Proceedings of the Ninth IEEE Conference on Artificial Intelligence for Applications.*, pp. 93-99. IEEE Computer Society Press.

Riloff, E. and W. Lehnert. (1992). "Classifying Texts Using Relevancy Signatures", in *Proceedings of the Tenth National Conference for Artificial Intelligence.* San Jose, CA. pp. 329-334.

Schank, R. (1991). "Where's the AI? in *AI Magazine*. Vol. 12, no. 4 - winter 1992. AAAI Press, Menlo Park, CA. pp. 38-49.

Sundheim, B. (1992). *Proceedings of the Fourth Message Understanding Conference.*

Tomkins, S. (1965). The psychology of being right -- and left. *Trans-action,* 3, 23-27.

Tukey, J.W. (1969). Analyzing data: Sanctification or detective work? *American Psychologist,* 24, 83-91.

Weizenbaum, J. (1976). *Computer Power and Human Reason*. W.H. Freeman.