ACL Lifetime Achievement Award

The Dawn of Statistical ASR and MT

Frederick Jelinek*
Johns Hopkins University

I am very grateful for the award you have bestowed on me. To understand your generosity I have to assume that you are honoring the leadership of three innovative groups that I headed in the last 47 years: at Cornell, IBM, and now at Johns Hopkins. You know my co-workers in the last two teams. The Cornell group was in Information Theory and included Toby Berger, Terrence Fine, and Neil J. A. Sloane (earlier my Ph.D. student), all of whom earned their own laurels.

I was told that I should give an acceptance speech and was furnished with example texts by previous recipients. They wrote about the development and impact of their ideas. So I will tell you about my beginnings and motivations and then focus on the contributions of my IBM team. In this way the text will have some historical value and may clear up certain widely held misconceptions.

1. Beginnings

Information Theory seemed to be one of the most prestigious disciplines during my years as a student at MIT (1954–1962). The faculty included the founders of the field—Shannon, Fano, Elias, and others. Some of my contemporaries were Viterbi, Jacobs, Kleinrock (founders of Qualcom), Gallagher, Kailath, and Massey. Not daring to approach Shannon himself, I asked Professor Fano to be my thesis adviser. I was making slow progress when in 1961, after three years of trying, I succeeded in extricating my future wife Milena from communist Czechoslovakia (how this was accomplished is another story) and married her. One problem we needed to solve was how she should occupy herself during the long hours I was spending in the underground stacks of the MIT library. At the time the famous linguist Roman Jacobson was simultaneously a University Professor at Harvard and an Institute Professor at MIT. Russian by origin, he spent 18 inter-war years (1920–1938) in Czechoslovakia, where he became one of the founders of the Prague Linguistic Circle. He had a Czech wife, the anthropologist Svatava Pirkova. He continued to maintain his connections with Czechs, and even young Czechs. I was invited to dinner at his house several times, once also with the newly arrived Milena. Jacobson was well known to have an eye for beautiful young women and he was reputed to enjoy exercising his influence. When my wife asked him for advice as to what to do, he suggested that she take up a fellowship at MIT which he would arrange for her to get. As promised, she got the fellowship and enrolled in the Ph.D. program of the Linguistics department. It should be appreciated that Jacobson did not interview her in a non-social setting and was aware that her previous schooling

* Center for Language and Speech Processing, Johns Hopkins University, 320 Barton Hall, 3400 N. Charles Street, Baltimore, MD 21218, USA. E-mail: jelinek@jhu.edu. This article is the text of the talk given on receipt of the ACL's Lifetime Achievement Award in 2009.

in Prague consisted only of one year of Slavic studies followed by two years at the Film Academy.

So Milena started attending lectures, several of them taught by Noam Chomsky. I sat in with her and got the crazy notion that I should switch from Information Theory to Linguistics. I went so far as to explore this notion with Professor Chomsky. Of course, word got around to my adviser Fano, whom it really upset. He declared that I could contemplate switching only after I had received my doctorate in Information Theory. I had no choice other than to obey. Soon thereafter, my thesis almost finished, I started interviewing at universities for a faculty position. After my job talk at Cornell I was approached by the eminent linguist Charles Hockett, who said that he hoped that I would accept the Cornell offer and help develop his ideas on how to apply Information Theory to Linguistics. That decided me. Surprisingly, when I took up my post in the fall of 1962, there was no sign of Hockett. After several months I summoned my courage and went to ask him when he wanted to start working with me. He answered that he was no longer interested, that he now concentrated on composing operas.

Discouraged a second time, I devoted the next ten years to Information Theory. This was the golden period of government support for science and technology. It seemed easy to get grants. Perhaps that was the reason I neglected to make any arrangements for work during the coming summer of 1972. I phoned Joe Raviv (whom I knew as a colleague from my sabbatical at IBM in 1968–69) to ask if I could spend three months in his group in Yorktown Heights. His answer was "Certainly, the sooner you arrive the better. We are starting to work on speech recognition."

By the time I arrived to take up that summer job, Raviv was promoted to manager of the IBM Scientific Center in Haifa, and IBM was negotiating with Professor Jonathan Allen of MIT to take over the speech group. These negotiations were not successful, and several weeks later the job was offered to me. I requested Cornell to grant me a leave of absence; they did, and I joined IBM (the following year I asked for and got another year, but when I tried to carry out the same maneuver in 1974, I was turned down).

Why did IBM start research in speech recognition? Believe it or not, IBM was worried that, with the advance of computing power, there might soon come a time when all the need for further improvements would disappear, and IBM business would dry up. Somebody came up with the suggestion that speech recognition would require lots of computing cycles. A task force was put together in 1971 to study the matter. The group included John Cocke (inventor of RISC machines), Herman Goldstine (right hand of von Neumann in research leading to ENIAC) and others. It recommended that a Continuous Speech Recognition group be established in the Research Division.

So the CSR group was started in early 1972 under the management of Joe Raviv. At the time IBM had a small speech group in one of its development laboratories in Raleigh, NC. (Actually, IBM "had" speech recognition even earlier. At the 1964 World's Fair in New York, Ernest Nassimbene demonstrated an isolated digits recognizer "in a shoe box.") Its three main members, Das, Dixon, and Tappert, were transferred from Raleigh to the Research Division in Yorktown. High management concluded that to get going the speech group would need the help of linguists. It transferred temporarily Fred Damerau, Stan Petrick, and Jane Robinson from linguistics to CSR. The staffing of the group was then completed by volunteers from the Computer Sciences Department: Lalit Bahl, Raimo Bakis, George Nagy, and others (later Jim and Janet Baker joined as well). But at the time only Bakis, Das, Dixon, and Tappert knew anything about speech. Towards the end of the summer I took over the direction of the group and received a gift from heaven: the freshly graduated physicist Robert Mercer, who in the spring accepted an IBM job in a group that was abolished before he arrived in September.

Table 1
Sentences from the Resource Management language.

Show locations and C-ratings for all deployed subs that were in their home ports April 5.
List the cruisers in Persian Sea that have casualty reports earlier than Jarrett's oldest one.
How many ships were in Galveston May 3rd?
Is Puffer's remaining fuel sufficient to arrive in port at the present speed?
How many long tuns is the average displacement of ships in Bering Strait?

2. The Competition

In 1971, parallel to the work of the IBM task force, ARPA established a project in Speech Understanding. I don't know what led to that decision, but the main forces behind it were Allen Newell and J. C. R. Licklinder. Funds were provided to Carnegie Mellon, Systems Development Corporation, Bolt Beranek & Newman, and probably SRI, Sperry-Univac, University of Pennsylvania, UC Berkeley, and UCLA. Not all of these institutions were to field complete systems. For instance, Ohio, UCLA, and Berkeley provided consulting by linguists (Peter Ladefoged, Vicky Fromkin, John Ohala, Michael O'Malley, and June Shoup).

Here are the names of some other researchers who attended the meetings organized by the new project: Raj Reddy, Dennis Klatt, L. D. Erman, V. Lesser, Bruce Lowerry, Bonnie Nash-Webber, George White, Fil Alleva, Wayne Ward, Don Walker, Victor Zue, Stephanie Sennef, Bill Woods, John Makhoul, Wayne Lea, Beatrice Oshika, and Janet and Jim Baker. IBM was invited to attend the meetings, but we did not compete. ARPA was a six-year project which was supposed to recognize (and interpret?) sentences from a "Resource Management" grammar; for an example of the sentences generated, see Table 1. At the end of the six-year period the project was declared a success because it "met its goals."

Clearly the best of the constructed recognizers was the Dragon System (Baker 1975) implemented by the Bakers, graduate students at CMU. It used Hidden Markov models (HMMs) whereas the rest of the ARPA participants based their work on templates, Dynamic Time Warping (DTW), and hand-written rules. The best of these latter systems was Harpy, developed by another CMU graduate student, Bruce Lowerry.

3. IBM's Initial Formulation

For our first task we decided to recognize sentences generated by the so-called New Raleigh grammar, a finite–state device whose schematic is shown in Figure 1. The grammar is actually a Hidden Markov Model. State transitions generate words and are taken with uniform probability. Generation starts in the initial state, a transition is taken, and a word from the list associated with that transition is selected with uniform probability; then one of the transitions out of the new state is taken (again with uniform probability), a word corresponding to that transition is again selected at random, a new state is reached, and so on. The process continues until one arrives at the final state. The grammar generates such bizarre sentences as are shown in Table 2.[1]

1 Note that, from a syntactic point of view, all the sentences generated are structurally English, even though the vast majority make no sense.

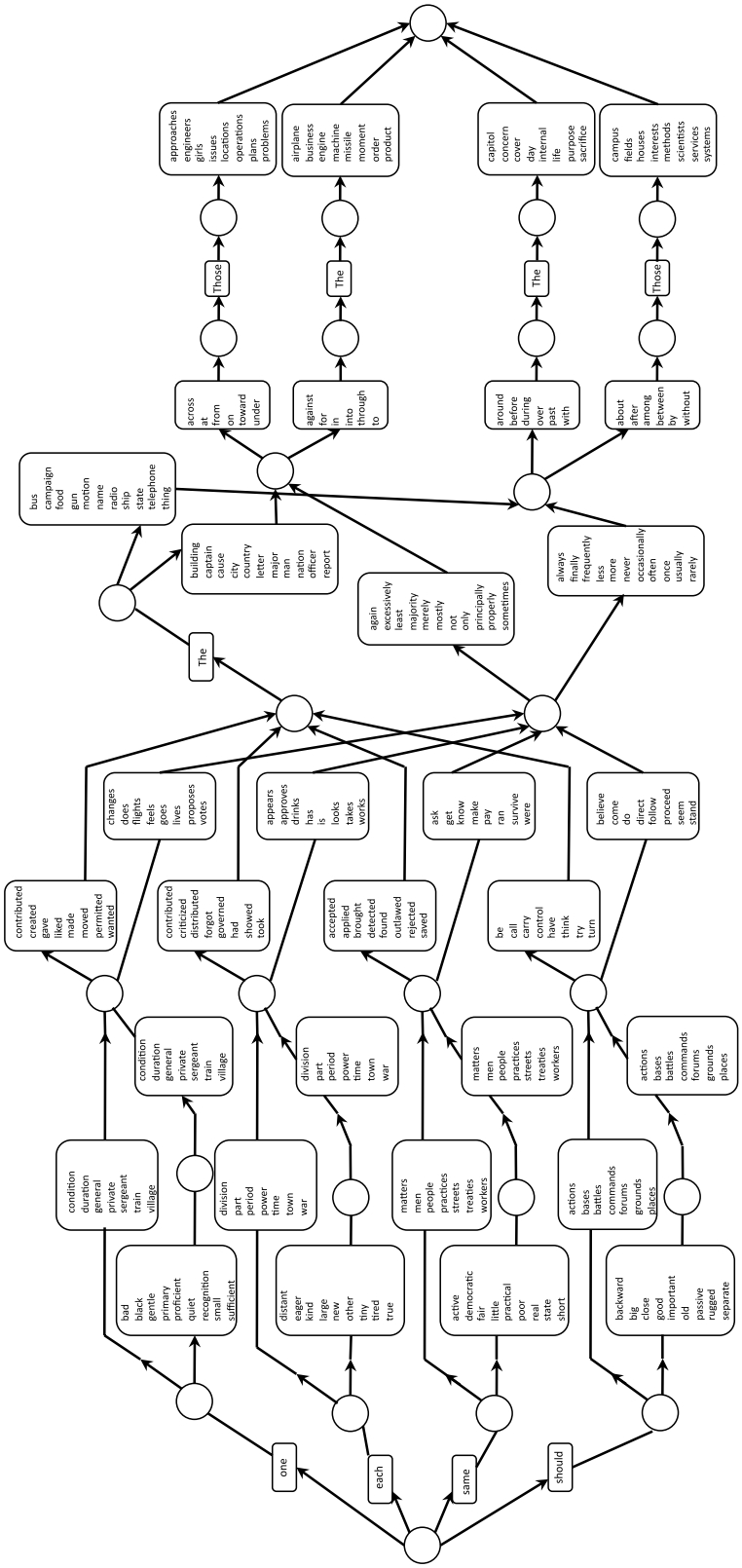

Figure 1
The New Raleigh grammar.

Table 2
Sentences generated by the New Raleigh grammar.

Each distant town disturbed the telephone during the purpose.
Some matters survive never between those systems.
Should important grounds be the radio over the concern?
Some matters rejected the radio during the capitol.
One recognition condition proposes only across those engineers.

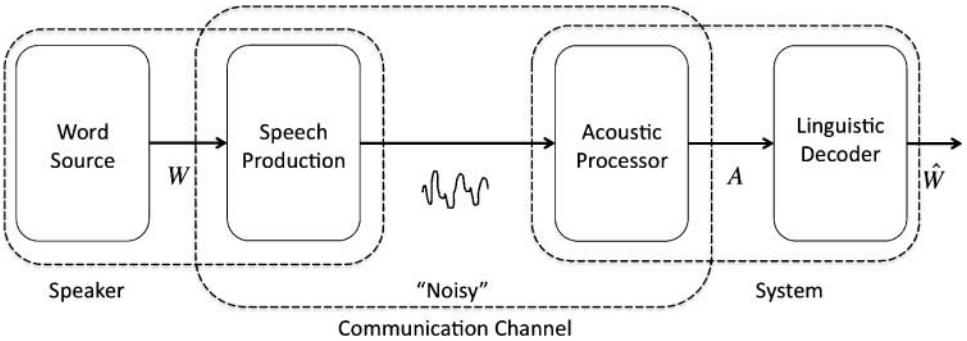

Figure 2
Schematic of statistical speech recognition.

The generated sentences were read (as naturally as possible) by native American speakers and were recorded in a special sound-proofed room. The recorded speech signal was transformed into a string $\mathbf{A}$ of symbols (one symbol per centisecond) chosen from an "alphabet" $\mathcal{A}$ of size 200. The alphabet $\mathcal{A}$ itself was extracted by **vector quantization** from a training sample of the speech signal.

Our formulation of the combined generation/recognition process is seen in Figure 2. The entire operation can be regarded as transmission through a noisy channel, the basic problem of Information Theory.[2] Its corresponding mathematical formula is

$$\widehat{\mathbf{W}} = \arg\max_{\mathbf{W}\in\mathcal{S}} P(\mathbf{W}|\mathbf{A}) = \arg\max_{\mathbf{W}\in\mathcal{S}} P(\mathbf{A}|\mathbf{W})\,P(\mathbf{W})$$

where $\mathbf{W}$ denotes a string of words, $\mathbf{A}$ denotes the acoustic signal observed by the recognizer, and $\mathcal{S}$ is the set of sentences that can be generated by the grammar. The formula calls for a statistical approach. The noisy channel of Figure 2 has input $\mathbf{W}$ and output $\mathbf{A}$ and is characterized by the probability $P(\mathbf{A}|\mathbf{W})$. The designer of the recognizer must attempt to estimate the channel probabilities $P(\mathbf{A}|\mathbf{W})$ and the a priori probability $P(\mathbf{W})$ that the speaker will utter the word string $\mathbf{W}$. Once the models $\widehat{P}(\mathbf{A}|\mathbf{W})$ and $\widehat{P}(\mathbf{W})$[3]

2 No wonder that we immediately hit upon the diagram: Both Lalit Bahl and I wrote Information Theoretic Ph.D. theses.

3 $\widehat{P}(A|W)$ is called the **acoustic model** and $\widehat{P}(W)$ is called the **language model.** Both these terms, now used throughout the field, were "invented" at IBM, as was the term **perplexity**, denoting an Information Theoretic measure of the difficulty of the recognition task.

are established, the recognizer when it observes **A** can conduct a search leading to the maximizing word string $\widehat{\mathbf{W}}$.

In the case of an artificial grammar, such as the New Raleigh grammar, the model $\widehat{P}(\mathbf{W})$ is in fact equal to the actual probability $P(\mathbf{W})$. Furthermore, because the set $\mathcal{S}$ of word strings **W** over which we maximize can be listed, the difficulty of the task can be measured approximately (because the acoustic similarity of words is not taken into account) by entropy:

$$H(\mathbf{W}) \triangleq -\sum_{\mathbf{W}\epsilon\mathcal{S}} P(\mathbf{W}) \log P(\mathbf{W})$$

However, because the participants in the ARPA project introduced the false measure "branching factor" (which was the **arithmetic** mean of the out-of-state branching of their finite–state grammar), we replaced $H(\mathbf{W})$ as a measure of difficulty by **perplexity**, defined by

$$PP \triangleq 2^{H(\mathbf{W})}$$

It turned out that the New Raleigh grammar had approximate perplexity 7 whereas the Resource Management grammar had 2.

The early IBM approach was described in three papers: Jelinek, Bahl, and Mercer (1975), Bahl and Jelinek (1975), and Jelinek (1976).

4. The Tangora

In 1978 (or so), we thought it was time to abandon artificial grammars and to start recognizing "natural" speech. We settled on a 5,000-word vocabulary and set ourselves the task of recognizing read sentences from an IBM internal correspondence corpus. Our ambition was to use a combination of IBM array processors to achieve essentially real-time performance. We fulfilled a promise to the management to achieve it by 1984. To make things easier on ourselves we limited the recognizer to the transcription of discrete speech, where sentences were spoken with pauses between words. We got rid of the sound room and recorded the readings on close-talking microphones. The system was to be speaker-sensitive, that is, acoustic models were trained for each individual reader separately.

It is worth noting that during this time we persuaded the IBM management to hire Jim and Janet Baker, who were not to receive their Ph.D.s from Carnegie Mellon until a year later. At the same time, "our" three linguist helpers returned to their original group.

When handling natural speech, the main question was how to estimate the language model $\widehat{P}(\mathbf{W})$. There was no simple way of achieving this. We thought that the right approach ought to be somehow related to English grammar. The linguist Stan Petrick, while he still was with us, said "Don't worry, I will just make a little grammar." Of course he never did, and the phrase acquired a mythical status in the manner of "famous last words."

So at John Cocke's suggestion we decided to model English by trigrams. That is, we used the approximation

$$\widehat{P}(\mathbf{W}) = P(w_1, w_2, w_3 ... w_k) \approx$$
$$\widehat{P}(w_1, w_2)\ \widehat{P}(w_3|w_1, w_2)\ \widehat{P}(w_4|w_2, w_3) .... \widehat{P}(w_k|w_{k-2}, w_{k-1})$$

But this decision did not dispose of the problem. How should we estimate the basic building blocks $P(w_k|w_{k-2}, w_{k-1})$? It was clear that the estimate would have to be based on trigram counts $C(w_{k-2}, w_{k-1}, w_k)$ assembled from some appropriate text (the one that provided us with the read speech). But relative frequencies

$$f(w_k|w_{k-2}, w_{k-1}) \triangleq \frac{C(w_{k-2}, w_{k-1}, w_k)}{C(w_{k-2}, w_{k-1})}$$

would not suffice. Indeed, the speech would frequently involve trigrams w_{k-2}, w_{k-1}, w_k whose count $C(w_{k-2}, w_{k-1}, w_k)$ in the training corpus equalled 0. Then if w_{k-2}, w_{k-1}, w_k were uttered by the speaker (reader), the recognizer would necessarily make an error. What was required was *smoothing*, and the type we chose at first was linear interpolation:

$$\widehat{P}(w_k|w_{k-2}, w_{k-1}) = \lambda_3 f(w_k|w_{k-2}, w_{k-1}) + \lambda_2 f(w_k|w_{k-1}) + \lambda_1 f(w_k)$$

where $\lambda_j's$ would be non-negative, would satisfy $\lambda_3 + \lambda_2 + \lambda_1 = 1$, and would be optimally chosen (we knew how).

Just as in the initial phase of IBM CSR research, the acoustics **A** input to the recognizer were a string of centisecond symbols chosen from an alphabet $\mathcal{A}$ which itself was derived by vector quantization. This discretization of speech allowed for an easier estimate of parameters defining the **acoustic processor** model $\widehat{P}(\mathbf{A}|\mathbf{W})$. The latter was implemented as a concatenation of HMMs, one for each word of the string **W**. An example of such an HMM (itself a concatenation of HMMs as prescribed by the pronunciation lexicon) is shown in Figure 3.

Having defined the signal processing leading to the string **A**, and the structures of the acoustic and language models $\widehat{P}(\mathbf{A}|\mathbf{W})$ and $\widehat{P}(\mathbf{W})$, it remains to discuss the search for the recognizer output $\widehat{\mathbf{W}}$ implicit in the formula

$$\widehat{\mathbf{W}} = \arg\max_{\mathbf{W}} \widehat{P}(\mathbf{A}|\mathbf{W})\ \widehat{P}(\mathbf{W})$$

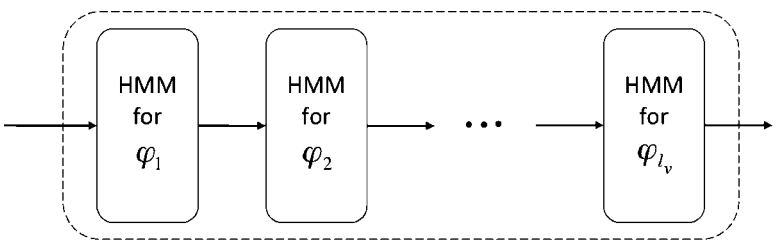

Figure 3
HMM for the word v realized by phone sequence $\varphi_1 \varphi_2 \cdots \varphi_{l_v}$.

We used the appropriately modified Viterbi algorithm version of dynamic programming (Viterbi 1967), a decision we took in spite of the fact that the algorithm would carry out a sub-optimal search.

We called the system **Tangora** after Albert Tangora who, during a 1923 business show in New York, ran off a total of 8,840 correctly spelled words in one hour of nonstop typing, a rate of 147 words per minute. Incredibly, it was estimated that Tangora executed an average of twelve-and-a-half strokes per second!

5. Some ASR Firsts

During my time at IBM, my colleagues and I pioneered various techniques that were later taken over by the entire field. Here are some examples:

- We replaced the discrete signal processing leading to **A** by directly modeling the real input as a mixture of Gaussians. Also, the resulting HMM now generated outputs from states instead of from transitions.
- We introduced pronunciation modeling by triphones, thereby modeling phones as influenced by context. That is, in our pronunciation lexicon each word was first transformed into a string of symbols roughly corresponding to phones. The HMM model of a word then became a concatenation of triphone models. Here is an example:

$$table \Longrightarrow T\ EI\ B\ L \Longrightarrow \#\ T\ EI \quad T\ EI\ B \quad EI\ B\ L \quad B\ L\ \#$$

- The phone alphabet was of the order of 50. That would necessitate $50^3 = 125,000$ models (one for each triphone), an intolerable number to use and/or estimate parameters for. To reduce this number meant categorizing the context into equivalence classes. This was accomplished by clustering carried out by **decision trees**.
- We tried to move from the simple trigram language model to a more sophisticated one. Slava Katz designed a back-off model (Katz 1987) relying on Good-Turing probability estimation. It was an improvement which moved the state of the art forward, but it was later superseded by Kneser-Ney smoothing. Because language modeling is a search for equivalence classification, we tried several methods based on decision trees. These were an improvement, but the attendant complication in implementation was not deemed justified, particularly now that enormous amounts of text are available from the Internet.
- Aware that a Viterbi search for the maximizing word string $\widehat{\mathbf{W}}$ is not optimal, Lalit Bahl developed a multi-stack search algorithm inspired by the stack algorithm of Information Theory (Jelinek 1969). The procedure was later formalized and cleaned up by Doug Paul of Lincoln Laboratories (Paul 1992). The multi-stack algorithm required the invention of a Fast Acoustic Match which suggested word candidates for detailed examination on the basis of a parallel, crude acoustic-fit test.
- Many problems of probability estimation involve simultaneous consideration of influence by disparate phenomena. We found that Maximum Entropy estimation is a method suitable for treating the

situation. This method facilitated the development by the summer intern Ronnie Rosenfeld of a Trigger Language Model that allowed the discourse topic to influence word prediction (Rosenfeld 1996).

- It was hoped that language modeling could be strengthened by including consideration of the parts-of-speech of words preceding the prediction. This unfortunately did not turn out to be the case. Nevertheless, Bahl and Mercer pioneered a method of part-of-speech tagging by HMM (Bahl and Mercer 1976) that is now widely used in various applications of Natural Language Processing.

6. IBM Influence on the Speech and Language Field

The success of the statistical formulation of the speech recognition problem led to invitations to share our methods with a wider audience. We presented several courses teaching our data-centric approach. In 1980 we gave a course in Udine, Italy, organized by CISM (International Centre for Mechanical Sciences); in 1983 a two-week course at MIT; and finally in 1986 a course in Oberlech, Austria, organized by IBM Scientific Centers. Furthermore, I was invited to give a keynote speech at the 1990 ACL Meeting in Pittsburgh, PA.

We were not satisfied with the crude *n*-gram language model we were using and were "sure" that an appropriate grammatical approach would be better. Because we wanted to stick to our data-centric philosophy, we thought that what was needed as training material was a large collection of parses of English sentences. We found out that researchers at the University of Lancaster had hand-constructed a "treebank" under the guidance of Professors Geoff Leach and Geoff Sampson (Garside, Leech, and Sampson 1987). Because we wanted more of this annotation, we commissioned Lancaster in 1987 to create a treebank for us. Our view was that what we needed above all was quantity, possibly at some expense of quality: We wanted to extract the grammatical language model statistically and so a large amount of data was required. Another belief of ours was that the parse annotation should be carried out by intelligent native speakers of English, not by linguistic experts, and that any inaccuracies would naturally cancel each other. Indeed, the work was done by Lancaster housewives led by a high school drop-out.

Meanwhile, Geoff Leech set out to assemble the British National Corpus and we thought that the United States should have something like that as well. So in 1987 I arranged to visit Jack Schwartz, who was the boss of Charles Wayne at DARPA, and I explained to him what was needed and why. He immediately entrusted Charles with the creation of the appropriate organization. One of the problems was where the eventual corpus should reside. Deep-pocketed IBM would be unsuitable: Possessors of desirable corpora would charge immoderate sums for the acquisition of rights. I thought that only a university would do. So I inquired of Aravind Joshi and Mitch Marcus (and perhaps even Mark Liberman) at the 1988 Conference of Applied Natural Language Processing in Austin whether the required site could be the University of Pennsylvania. My colleagues were interested, and Charles Wayne invited appropriate people to a meeting at the Lake Mohunk Mountain House to discuss the matter. That is how the Linguistic Data Consortium was born.

Once the commissioned treebank was delivered, we started to experiment with parsers. Jointly with the University of Pennsylvania we applied for and received an

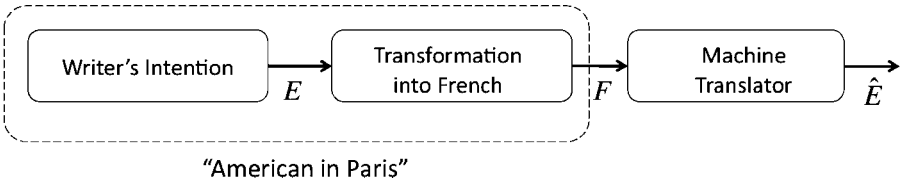

Figure 4
Schematic of statistical machine translation.

NSF grant for grammatical development. The eventual result was SPATTER, implemented by David Magerman. It used statistics, decision trees, and a history-based operation. The University of Pennsylvania graduate student Eric Brill developed his transformation-based learning (Brill 1992) that he applied to part-of-speech tagging and grammar derivation. Ezra Black of IBM organized a committee whose aim was the specification of a metric suitable for evaluation of parser performance. The result was the PARSEVAL measure.

7. Machine Translation

Even though the problem of speech recognition remains unsolved to this day, some of us started to wonder in the mid 1980s whether our ASR methods could be successfully applied to new fields. Bob Mercer and I spent many of our after-lunch "periphery" walks discussing possible candidates. We soon came up with two: machine translation and stock market modeling. It is probably only coincidence that Bob eventually ended up investigating the possibilities of stock value prediction. Indeed, he and Peter Brown departed IBM in 1993 to work for the phenomenally successful hedge fund Renaissance Technologies. Eventually at least 10 former members of the IBM CSR group were to be employed by that same company. The performance of the Renaissance fund is legendary, but I have no idea whether any methods we pioneered at IBM have ever been used. My former colleagues will not tell me: Theirs is a very hush-hush operation!

On the other hand, we did start working on machine translation (MT) in 1987. As expected, we formulated the problem statistically. The basic diagram of MT, shown in Figure 4, is practically identical to that of ASR. In fact, even the basic formulas are identical except for a change in the letters that designate the variables:

$$\widehat{E} = \arg\max_{E} P(F|E)\, P(E)$$

This formula assumes translation from the foreign language[4] F into English E. The formulation pretends that the speaker's mind works in English, the generated text is then translated by him into the foreign language F, and the task of the machine translator is to ferret out the speaker's original thought $\widehat{E}$.

That we wanted to translate into English was a given—not because of the utility of the task, but because our knowledge of English would allow us to judge the quality

4 Originally F stood for French.

of the translation. We wanted to make the problem real, yet as easy as possible. So we looked for a language *F* that was relatively close to English. The answer was French. Because we wanted the process to be data-centric, we searched for a pair of corpora *F* and *E* that would be translations of each other. Luck was with us: The Canadian parliament Hansards (proceedings) were maintained in English and French.[5] So statistical language translation was born (Brown et al. 1990), and the descendants of our original methods are being continually improved.

When we started, none of us spoke French, so we decided to learn it. We uncovered a small institute whose location was opposite New York Grand Central Station on 42nd Street. The institute advertised that it would teach French to anybody in two (!) weeks of intensive immersion. We didn't believe it, of course, but because the costs and location were convenient, we started on our daily commute. I will not go into the semi-fraudulent aspects of the operation, but Lalit Bahl, Peter Brown, Bob Mercer, and I had a lot of fun and did advance considerably our knowledge of French.

When we had our first results we submitted them to Coling 1988. Here is a part of the rejection review we received:

> The validity of a statistical (information theoretic) approach to MT has indeed been recognized, as the authors mention, by Weaver as early as 1949. And was universally recognized as mistaken by 1950 (cf. Hutchins, MT – Past, Present, Future, Ellis Horwood, 1986, p. 30ff and references therein). The crude force of computers is not science. The paper is simply beyond the scope of COLING.
>
> Anonymous Coling review, 1 March 1988

8. Conclusion

Research in both ASR and MT continues. The statistical approach is clearly dominant. The knowledge of linguists is added wherever it fits. And although we have made significant progress, we are very far from solving the problems. That is a good thing: We can continue accepting new students into our field without any worry that they will have to search, in the middle of their careers, for new fields of action.

References

Bahl, L. R. and F. Jelinek. 1975. Decoding for channels with insertions, deletions, and substitutions with applications to speech recognition. *IEEE Transactions on Information Theory*, IT-21:404–411.

Bahl, L. R. and R. L. Mercer. 1976. Part of speech assignment by a statistical algorithm. In *IEEE International Symposium on Information Theory*, pp. 88–89, Ronneby, June.

Baker, J. K. 1975. The dragon system: an overview. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-23(1):24–29.

Brill, E. 1992. A simple rule-based part of speech tagger. In *Proceedings of the Third Conference on Applied Natural Language Processing*, pages 152–155, Trento, Italy. ACL.

Brown, P. F., J. Cocke, S. A. DellaPietra, V. DellaPietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. Roosin. 1990. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85.

Garside, R., G. Leech, and G. Sampson. 1987. *Computational Analysis of English: A Corpus-Based Approach*. Longman, London.

5 Every sentence spoken by a deputy in one language was faithfully translated into the other.

Jelinek, F. 1969. A fast sequential decoding algorithm using a stack. *IBM Journal of Research and Development*, 13:675–685.

Jelinek, F. 1976. Continuous speech recognition by statistical methods. *IEEE Proceedings*, 64(4):532–556.

Jelinek, F., L. R. Bahl, and R. L. Mercer. 1975. Design of a linguistic statistical decoder for the recognition of continuous speech. *IEEE Transactions on Information Theory*, IT-21:250–256.

Katz, S. 1987. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-35(3):400–401.

Paul, D. B. 1992. An essential a^* stack decoder algorithm for continuous speech recognition with a stochastic language model. In *Proceedings of the 1992 International Conference on Acoustics, Speech, and Signal Processing*, pages 25–28, San Francisco.

Rosenfeld, R. 1996. A maximum entropy approach to adaptive statistical language modeling. *Computer Speech and Language*, 10(3):187–228.

Viterbi, A. J. 1967. Error bounds for convolutional codes and an asymmetrically optimum decoding algorithm. *IEEE Transactions on Information Theory*, IT-13:260–267.