7

Natural Language Processing

BEYOND PATTERN RECOGNITION OF INDIVIDUAL ALPHANUMERIC CHARACTERS, whether they be of fixed font or handwritten, lies the problem of understanding strings of characters that form words, sentences, or larger assemblages of text in a "natural" language, such as English. To distinguish languages such as English from the languages used by computers, the former are usually called "natural languages." In artificial intelligence, "understanding" natural language input usually means either converting it to some kind of memory model (such as the one used by Raphael in his SIR system or the semantic network used by Quillian) or the evocation of some action appropriate to the input.

Natural languages are spoken as well as written. And, because speech sounds are not as well segmented as are the characters printed on a page, speech understanding presents additional difficulties, which I'll describe in a later chapter.

The inverse of understanding natural language input is generating natural language output – both written and spoken. Translating from one language to another involves both understanding and generation. So does carrying on a conversation. All of these problems – understanding, generation, translation, and conversing – fall under the general heading of "natural language processing" (sometimes abbreviated as NLP).

7.1 Linguistic Levels

Linguists and others who study language recognize several levels at which language can be analyzed. These levels can be arranged in a sort of hierarchy, starting with those dealing with the most basic components of language (sounds and word parts) and proceeding upward to levels dealing with sequences of sentences. If speech is being dealt with, there are the levels of *phonetics* (language sounds) and *phonology* (organization of sounds into words). For both speech and text, *morphology* deals with how whole words are put together from smaller parts. For example, "walking" consists of "walk" plus "-ing."

Next, *syntax* is concerned with sentence structure and grammar. It attempts to describe rules by which a string of words in a certain language can be labeled either grammatical or not. For example, the string "John hit the ball" is grammatical but the string "ball the hit John" is not. Together with the dictionary definitions of words, syntax comes next in importance for understanding the meaning of a sentence. For example, the sentence "John saw the man with a telescope" has two different meanings depending on its syntactic structure (that is, depending on whether "with a telescope" refers to "the man" who had a telescope or to "saw").

But grammaticality alone is insufficient for determining meaning. For example, the sentence "Colorless green ideas sleep furiously" might be considered grammatical, but it is nonsensical. The *semantics* level helps to determine the meaning (or the meaninglessness) of a sentence by employing logical analyses. For example, through semantic analysis, an "idea" can't be both "colorless" and "green."

Next comes the *pragmatics* level, which considers the context of a sentence to pin down meaning. For example, "John went to the bank" would have a different meaning in a sentence about stream fishing than it would in a sentence about commerce. Pragmatics deals with meanings in the context of specific situations.

One of these levels in particular, namely, syntax, was the subject of much early study and continues to be an important aspect of NLP. In 1957, the American linguist Noam Chomsky published a ground-breaking book titled *Syntactic Structures* in which he proposed sets of grammatical rules that could be used for generating the "legal" sentences of a language.[1] The same rules could also be used to analyze a string of words to determine whether or not they formed a legal sentence of the language. I'll illustrate how this analysis is done using what Chomsky called a *phrase-structure grammar* (PSG).[2] The process is very similar to how we all "diagrammed" sentences back in grade school.

Grammars are defined by stating rules for replacing words in the string by symbols corresponding to syntactic categories, such as noun or verb or adjective. Grammars also have rules for replacing strings of these syntactic symbols by additional symbols. To illustrate these ideas, I'll use a very simple grammar adapted from one of Chomsky's examples. This grammar has only three syntactic categories: determiner, noun, and verb. Those three are sufficient for analysing strings such as "the man hit the ball."

One of the rules in this illustrative grammar states that we can replace either of the words "the" or "a" by the symbol "DET" (for determiner). Linguists write this rule as follows:

> the | a → DET
> (The symbol | is used to indicate that *either* of the words that surround it can be replaced by the syntactic symbol to the right of the arrow.)

Here are some other rules, written in the same format:

> man | ball | john → N
> (The words "man," "ball," and "john" can be replaced by the symbol "N" for noun.)
>
> hit | took | threw → V
> (The words "hit," "took," and "threw" can be replaced by the symbol "V" for verb.)
>
> DET N → NP
> (The string of symbols "DET" and "N" can be replaced by the symbol "NP" for noun phrase.)

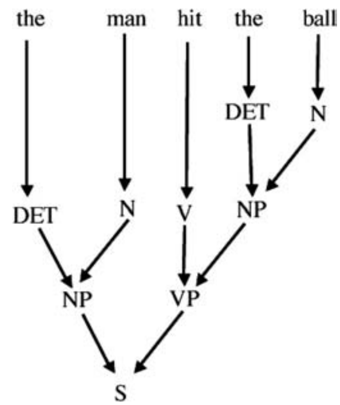

Figure 7.1. A parse tree for analyzing a sentence.

V NP → VP
(The string of symbols "V" and "NP" can be replaced by the symbol "VP" for verb phrase.)

NP VP → S
(The string of symbols "NP" and "VP" can be replaced by the symbol "S" for sentence.)

Symbols such as "S," "DET," "NP," and so on are called the "nonterminal" symbols of the language defined by the grammar, whereas vocabulary words such as "ball," "john," and "threw" are the "terminal" symbols of the language.

We can apply these rules to the string "the man hit the ball" to transform it into "S." Any string that can be changed into "S" in this way is said to be grammatical – a legal sentence in the language defined by this very simple grammar. One way to illustrate the rule applications, called a *parse tree*, is shown in Fig. 7.1.[3]

This example was based on a small set of syntactic categories and replacement rules just to illustrate the main ideas about syntactic analysis. To make the grammar slightly more realistic, we would need to include symbols and replacement rules for adjectives, adverbs, prepositions, and so on. And, of course, we would have to include many more vocabulary words.

Grammars are called *context-free grammars* (CFGs) if all of their rules have just a single nonterminal symbol on the right side of the arrow. They are called that because when the rules are used in reverse (to generate rather than to analyze grammatical sentences), the way in which a nonterminal symbol is replaced does not depend on the presence of any other symbols. PSGs are context free.

The diagram in Fig. 7.2 shows how the rules of our simple grammar can be used to generate sentences. In this case, it starts with the symbol for sentence, namely, "S," and generates the sentence "John threw the ball."

This simple grammar certainly can't generate all of the sentences we would claim to be legal or acceptable. It also generates sentences that we would not ordinarily want to accept, such as "the john threw the ball." Chomsky's book presents much more complex grammars, and later work has produced quite elaborate ones. By the early 1960s, several grammars had been encoded in computer programs that

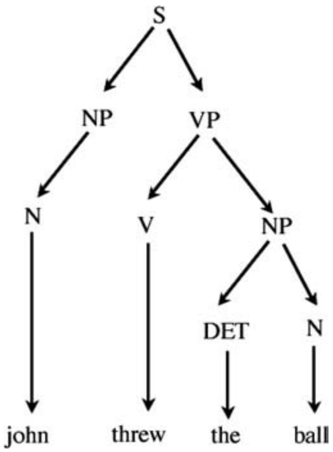

Figure 7.2. A parse tree for generating a sentence.

could parse samples of English text.[4] I'll be mentioning several different grammars, some more complex than CFGs in succeeding chapters. Nevertheless, even the most complex grammars can't cleanly distinguish between sentences we would accept as grammatically correct and those we would not. I will return to this difficulty and one way to deal with it in a later chapter.

The way a sentence is parsed by a grammar can determine its meaning, so an important part of natural language processing involves using the grammar rules to find acceptable parse trees for sentences. Finding a parse tree involves search – either for the several different ways that the nonterminal symbols, beginning with "S," can be replaced using grammar rules in an attempt to match a target sentence or for the several different ways the words in a target sentence can be replaced by nonterminal symbols in an attempt to produce the symbol "S." The first of these kinds of searches is called "top-down" (from "S" to a sentence); the second is called "bottom-up" (from a sentence to "S").

It is often (if not usually) the case that, given a grammar, sentences can have more than one parse tree, each with a different meaning. For example, "the man hit the ball in the park" could have a parse tree in which "in the park" is part of a verb phrase along with "hit" or a parse tree in which "in the park" is part of a noun phrase along with "ball." Moreover, as I have already mentioned, some parsings of sentences might be meaningless. For example, according to my simple grammar, "the ball threw the man" is a legal but probably meaningless sentence. Deciding which parse tree is appropriate is part of the process of deciding on meaning and is a job for the semantics (and possibly even the pragmatics) level. During the late 1950s and throughout most of the 1960s and beyond, syntactic analysis was more highly developed than was semantics.

Semantic analysis usually involves using the parse tree to guide the transformation of the input sentence into an expression in some well-defined "meaning representation language" or into a program that responds in the appropriate way to the input sentence. For example, "the man threw the ball" might be transformed into a logical expression such as

$$(\exists x, y, z)[\texttt{Past}(z) \wedge \texttt{Man}(x, z) \wedge \texttt{Ball}(y, z) \wedge \texttt{Event}(z) \wedge \texttt{Throws}(x, y, z)],$$

https://doi.org/10.1017/CBO9780511819346.009 Published online by Cambridge University Press

which can be interpreted as "there are x, y, and z, such that z is an event that occurred in the past, x is a man in that event, y is a ball in that event, and x throws y in that event."

Semantic analysis might also transform the sentence "the man threw the ball" into a program that, in some way, simulates a man throwing the ball in the past.

7.2 Machine Translation

Some of the first attempts to use computers for more than the usual numerical calculations were in automatic translation of sentences in one language into sentences of another. Word dictionaries could be stored in computer memory (either on tapes or on punched cards), and these could be used to find English equivalents for foreign words. It was thought that selecting an appropriate equivalent for each foreign word in a sentence, together with a modest amount of syntactic analysis, could be used to translate a sentence in a foreign language (Russian, for example) into English.

Reporting about a new computer[5] being developed by a team led by Harry D. Huskey at the National Bureau of Standards (now called the National Institute of Standards and Technology), the *New York Times* reported the following on May 31, 1949:[6]

> A new type of "electric brain" calculating machine capable not only of performing complex mathematical problems but even of translating foreign languages, is under construction here at the United States Bureau of Standards Laboratory at the University of California's Institute of Numerical Analysis. While the exact scope the machine will have in the translating field has not been decided, the scientists working on it say it would be quite possible to make it encompass the 60,000 words of the Webster Collegiate Dictionary with equivalents for each word in as many as three foreign languages.

Explaining how the machine might do translation, the *Times* reporter wrote

> When a foreign word for translation is fed into the machine, in the form of an electro-mathematical symbol on a tape or card, the machine will run through its "memory" and if it finds that symbol as record, will automatically emit a predetermined equivalent – the English word.
>
> . . .
>
> This admittedly will amount to a crude word-for-word translation, lacking syntax, but will nevertheless be extremely valuable, the designers say, for such purposes as scientists' translations of foreign technical papers in which vocabulary is far more of a problem than syntax.

The machine had not actually performed any translations – the idea of doing so was still just a possibility envisioned by Huskey. But even nonscientists could imagine the difficulties. An editorial in the *New York Times* the next day put the problem well:

> We have our misgivings about the accuracy of every translation. How is the machine to decide if the French word "pont" is to be translated as "bridge" or "deck" or to know that "operation" in German means a surgical operation? All the machine can do is to simplify the task of looking up words in a dictionary and setting down their English equivalents on a tape, so that the translator still has to frame the proper sentences and give the words their contextual meaning.

In a 1947 letter to Norbert Wiener, Warren Weaver, a mathematician and science administrator, mentioned the possibility of using digital computers to translate documents between natural human languages. Wiener was doubtful about this possibility. In his reply to Weaver, Wiener wrote "I frankly am afraid the boundaries of words in different languages are too vague and the emotional and international connotations are too extensive to make any quasi-mechanical translation scheme very hopeful." Nevertheless, by July 1949, Weaver had elaborated his ideas into a memorandum, titled "Translation" that he sent to several colleagues.

Weaver began his memorandum by stating the following:

> There is no need to do more than mention the obvious fact that a multiplicity of languages impedes cultural interchange between the peoples of the earth, and is a serious deterrent to international understanding. The present memorandum, assuming the validity and importance of this fact, contains some comments and suggestions bearing on the possibility of contributing at least something to the solution of the world-wide translation problem through the use of electronic computers of great capacity, flexibility, and speed.

According to the editors of the published volume[7] in which the memorandum was reprinted, "When he sent it to some 200 of his acquaintances in various fields, it was literally the first suggestion that most had ever seen that language translation by computer techniques might be possible." Weaver's document is often credited with initiating the field of machine translation (often abbreviated as MT).[8]

In June 1952 at MIT, Yehoshua Bar-Hillel (1915–1975), an Israeli logician who was then at MIT's Research Laboratory for Electronics, organized the first conference devoted to machine translation.[9] Originally optimistic about the possibilities, Bar-Hillel was later to conclude that full automatic translation was impossible.

In January 1954, automatic translation of samples of Russian text to English was demonstrated at IBM World Headquarters, 57th Street and Madison Avenue, New York City. The demonstration, using a small vocabulary and limited grammar, was the result of a collaboration between IBM and Georgetown University. The project was headed by Cuthbert Hurd, director of the Applied Sciences Division at IBM, and Léon Dostert of Georgetown. According to an IBM press release[10] on January 8, 1954,

> Russian was translated into English by an electronic "brain" today for the first time.
>
> Brief statements about politics, law, mathematics, chemistry, metallurgy, communications and military affairs were submitted in Russian by linguists of the Georgetown University Institute of Languages and Linguistics to the famous 701 computer of the International Business Machines Corporation. And the giant computer, within a few seconds, turned the sentences into easily readable English.
>
> A girl who didn't understand a word of the language of the Soviets punched out the Russian messages on IBM cards. The "brain" dashed off its English translations on an automatic printer at the breakneck speed of two and a half lines per second.
>
> "Mi pyeryedayem mislyi posryedstvom ryechyi," the girl punched. And the 701 responded: "We transmit thoughts by means of speech."
>
> "Vyelyichyina ugla opryedyelyayetsya otnoshyenyiyem dlyini dugi k radyiusu," the punch rattled. The "brain" came back: "Magnitude of angle is determined by the relation of length of arc to radius."

Although the demonstration caused a great deal of excitement and led to increased funding for translation research, subsequent work in the field was disappointing.[11] Evaluating MT work in a 1959 report circulated among researchers, Bar-Hillel had become convinced that fully automatic, high-quality translation (which he dubbed FAHQT) was not feasible "not only in the near future but altogether." His expanded report appeared in a 1960 paper that enjoyed wide distribution.[12]

One of the factors leading Bar-Hillel to his negative conclusions was the apparent difficulty of giving computers the "world knowledge" they would need for high-quality translation. He illustrated the problem with the following story:

> Little John was looking for his toy box. Finally he found it. The box was in the pen. John was very happy.

How should one translate "The box was in the pen"? Bar-Hillel argued that even if there were only two definitions of "pen" (a writing utensil and an enclosure where small children play), a computer knowing only those definitions would have no way of deciding which meaning was intended. In addition to its knowledge of vocabulary and syntax, a translating computer would need to know "the relative sizes of pens, in the sense of writing implements, toy boxes, and pens, in the sense of playpens." Such knowledge, Bar-Hillel claimed, was not at the disposal of the electronic computer. He said that giving a computer such encyclopedic knowledge was "utterly chimerical and hardly deserves any further discussion."

As later researchers would finally concede, Bar-Hillel was right about his claim that highly competent natural language processing systems (indeed, broadly competent AI systems in general) would need to have encyclopedic knowledge. However, most AI researchers would disagree with him about the futility of attempting to give computers the required encyclopedic knowledge. Bar-Hillel was well known for being a bit of a nay-sayer regarding artificial intelligence. (Commenting on John McCarthy's "Programs with Common Sense" paper at the 1958 Teddington Conference, Bar-Hillel said "Dr. McCarthy's paper belongs in the Journal of Half-Baked Ideas, the creation of which was recently proposed by Dr. I. J. Good.")[13]

In April 1964, the National Academy of Sciences formed the Automatic Language Processing Advisory Committee (ALPAC), with John R. Pierce (1910–2002) of Bell Laboratories as chair, to "advise the Department of Defense, the Central Intelligence Agency, and the National Science Foundation on research and development in the general field of mechanical translation of foreign languages." The committee issued its report in August 1965 and concluded, among other things, that ". . . there is no immediate or predictable prospect of useful machine translation."[14] They recommended support for basic linguistics science and for "aids" to translation, but not for further support of fully automatic translation. This report caused a dramatic reduction of large-scale funding of research on machine translation. Nonetheless, machine translation survived and eventually thrived, as we shall see in later chapters.

The Association for Machine Translation and Computational Linguistics (AMTCL) held its first meeting in 1962. In 1968, it changed its name to the Association for Computational Linguistics (ACL) and has become an international scientific and professional society for people working on problems involving natural language and computation. It publishes the quarterly journal *Computational Linguistics* and sponsors conferences and workshops.[15]

7.3 Question Answering

In addition to work on machine translation, researchers began exploring how sentences in a natural language, such as English, could be used to communicate with computers. You will recall Weizenbaum's ELIZA program that was able to engage a person in a conversation even though the program "understood" nothing about what was being said. And, I have already mentioned Raphael's SIR system that could represent information given to it and then answer questions.

I'll mention a few other projects to give a flavor of natural language processing work during this period. A program called BASEBALL (written in IPL-V, a special list-processing programming language developed by Newell, Shaw, and Simon to be described later) was developed at the Lincoln Laboratory under the direction of Bert Green, a professor of Psychology at the Carnegie Institute of Technology.[16] It could answer simple English questions about baseball using a database about baseball games played in the American League during a single year. For example, it could answer a question such as "Where did the Red Sox play on July 7?" The questions had to be of a particularly simple form and restricted to words in the program's vocabulary. In the authors' words,[17]

> Questions are limited to a single clause; by prohibiting structures with dependent clauses the syntactic analysis is considerably simplified. Logical connectives, such as *and*, *or*, and *not*, are prohibited, as are constructions implying relations like *most* and *highest*. Finally, questions involving sequential facts, such as "Did the Red Sox ever win six games *in a row*?" are prohibited.

The program worked by converting a question into a special form called a "specification list" using both special-purpose syntactic and semantic analyses. This list would then be used to access the program's database to find an answer to the question. For example, the question "Where did the Red Sox play on July 7?" would first be converted to the list:

Place = ?
Team = Red Sox
Month = July
Day = 7

The authors claimed that their "restrictions were temporary expedients that will be removed in later versions of the program." As far as I know, there were no later versions of the program. (As we will see as my history of AI unfolds, there are several instances in which it proved very difficult to remove "temporary" restrictions.)

Another natural language program, SAD SAM, was written in IPL-V in 1962–1963 by Robert Lindsay at the Carnegie Institute of Technology.[18] It could analyze English sentences about family relationships and encode these relationships in a family tree. Using the tree, it could then answer English questions about relationships.

https://doi.org/10.1017/CBO9780511819346.009 Published online by Cambridge University Press

For example, if SAD SAM received the sentence “Joe and Jane are Tom’s offspring,” it would construct a treelike list structure for a certain “family unit” in which Tom is the father and Joe and Jane are the children. Then, if it received the sentence “Mary is Jane’s mother,” it would add Mary to this structure as Tom’s wife. It would then be able to answer the question “Who is Joe’s mother?”

SAD SAM is an acronym for Sentence Appraiser and Diagrammer and Semantic Analyzing Machine. The SAD part parsed the input sentences and passed them to SAM, which extracted the semantic information needed for building family trees and for finding answers to questions. The program could accept a wide variety of sentences in Basic English – a system of grammar and a vocabulary of about 850 words defined by Charles K. Ogden.[19]

Robert F. Simmons (1925–1994), a psychologist and linguist at the Systems Development Corporation (SDC) in Santa Monica, California, had grander goals for his own work in natural language processing. According to an “In Memoriam” page written by Gordon Novak, one of his Ph.D. students at the University of Texas in Austin where Simmons took up a position as Professor of Computer Sciences and Psychology,[20]

> Simmons’ dream was that one could have “a conversation with a book;” the computer would read the book, and then the user could have a conversation with the computer, asking questions to be answered from the computer’s understanding of the book.

Accomplishing this “dream” would turn out to be as hard as AI itself. In a 1961 note about his proposed “Synthex” project, Simmons described how he would begin:[21]

> The objective of this project is to develop a research methodology and a vehicle for the design and construction of a general purpose computerized system for synthesizing complex human cognitive functions. The initial vehicle, proto-synthex, will be an elementary language-processing device which reads simple printed material and answers simple questions phrased in elementary English.

By 1965, Simmons and Lauren Doyle had conducted some experiments with their Protosynthex system. According to a report by Trudi Bellardo Hahn,[22] “A small prototype full-text database of chapters from a child’s encyclopedia (*Golden Book*) was loaded on the system. Protosynthex could respond to simple questions in English with an ‘answer.’ . . . it was a pioneering effort in the use of natural language for text retrieval.”

In the meantime, Daniel G. Bobrow (1935–), a Ph.D. student of Marvin Minsky’s at MIT, wrote a set of programs, called the STUDENT system, that could solve algebra “story problems” given to it in a restricted subset of English. Here is an example of a problem STUDENT could solve:

> The distance from New York to Los Angeles is 3000 miles. If the average speed of a jet plane is 600 miles per hour, find the time it takes to travel from New York to Los Angeles by jet.

STUDENT solved the problem by using some known relationships about speed and distance to set up and solve the appropriate equations. Bobrow’s dissertation gave several other examples of problems STUDENT could solve and the methods used.[23]

Notes

1. Noam Chomsky, *Syntactic Structures,* 's-Gravenhage: Mouton & Co., 1957. [104]
2. The basic structure of PSGs was independently invented by computer scientist John BackusBackus, John to describe the syntax of the ALGOL programming language. See John Backus, "The Syntax and Semantics of the Proposed International Algebraic Language of the Zürich ACM-GAMM Conference," *Proceedings on the International Conference on Information Processing*, pp. 125–132, UNESCO, 1959. [104]
3. According to C. George Boeree (see http://www.ship.edu/~cgboeree/wundtjames.html), Wilhelm Wundt "invented the tree diagram of syntax we are all familiar with in linguistics texts." [105]
4. For a survey of work during this period, see Daniel Bobrow, "Syntactic Analysis of English by Computer: A Survey," *Proceedings of the 1963 Fall Joint Computer Conference*, Vol. 24, pp. 365–387, Baltimore: Spartan Books, 1963. [106]
5. The Standards Western Automatic Computer (later abbreviated to SWAC) [107]
6. The quotation appears in John Hutchins, "From First Conception to First Demonstration: The Nascent Years of Machine Translation, 1947–1954. A chronology," *Machine Translation*, Vol. 12 No. 3, pp. 195–252, 1997. (A corrected 2005 version, with minor additions, appears at http://www.hutchinsweb.me.uk/MTJ-1997-corr.pdf.) [107]
7. W. N. Locke and A. D. Booth (eds.), *Machine Translation of Languages: Fourteen Essays*, pp. 15–23, Cambridge, MA: MIT Press, 1955. [108]
8. For a history of MT, see W. John Hutchins, "Machine Translation: A Brief History," in E. F. K. Koerner and R. E. Asher (eds.), *Concise History of the Language Sciences: From the Sumerians to the Cognitivists*, pp. 431–445, Oxford: Pergamon Press, 1995. (Also available online at http://www.hutchinsweb.me.uk/ConcHistoryLangSci-1995.pdf.) Hutchins also has a Web page devoted to his publicationsathttp://www.hutchinsweb.me.uk/. [108]
9. For reports about this conference see, E. Reifler, "The First Conference on Mechanical Translation," *Mechanical Translation*, Vol. 1 No. 2, pp. 23–32, 1954, and A. C. Reynolds, "The Conference on Mechanical Translation Held at MIT, June 17–20, 1952," *Mechanical Translation*, Vol. 1, No. 3, pp. 47–55, 1954. [108]
10. http://www-03.ibm.com/ibm/history/exhibits/701/701_translator.html. [108]
11. For a summary of the IBM–Georgetown work, see W. John Hutchins, "The Georgetown–IBMmachine translation!the Georgetown-IBM experiment Experiment Demonstrated in January 1954," in Robert E. Frederking and Kathryn B.Taylor (eds.), *Proceedings of Machine Translation: From Real Users to Research*, 6th Conference of the Association for Machine Translation in the Americas, AMTA-2004, pp. 102–114, Washington DC, USA, September 28–October 2, 2004, Berlin: Springer, 2004. An online version is available at http://www.hutchinsweb.me.uk/ATMA-2004.pdf. [109]
12. Yehoshua Bar-Hillel, "The Present Status of Automatic Translation of Languages," *Advances in Computers*, Vol. 1, No. 1, pp. 91–163, 1960. [109]
13. In D. V. Blake and A. M. Uttley (eds.), *Proceedings of the Symposium on Mechanisation of Thought Processes*, p. 85, London: Her Majesty's Stationary Office, 1959. [109]
14. John R. Pierce *et al.*, *Language and Machines: Computers in Translation and Linguistics*, ALPAC Report, National Academy of Sciences Publication 416, National Research Council, Washington, DC, 1966. [109]
15. See http://www.aclweb.org/. [110]
16. Bert F. Green Jr., Alice K. Wolf, Carol Chomsky, and Kenneth Laughery, "BASEBALL: An Automatic Question Answerer," pp. 219–224, *Proceedings of the Western Joint Computer Conference*, May 1961. Reprinted in Edward A. Feigenbaum and Julian Feldman (eds.),

Computers and Thought, pp. 207–216, New York: McGraw Hill, 1963, and in B. Grosz, K. Spark Jones, and B. Lynn Webber (eds.), *Readings in Natural Language Processing*, Morgan Kaufman, Los Altos, CA, 1986. [110]

17. *Ibid*. [110]
18. See Robert K. Lindsay, "Inferential Memory as the Basis of Machines Which Understand Natural Language," in Edward A. Feigenbaum, and Julian Feldman, *op. cit.*, pp. 217–233. [110]
19. Charles K. Ogden, *Basic English: A General Introduction with Rules and Grammar*, 4th edition, London: Kegan, Paul, Trench, Trubner & Co., Ltd., 1933. (Lindsay says 1,700 words; other sources say 850.) [111]
20. From http://www.cs.utexas.edu/users/ai-lab/simmons.html. [111]
21. Robert F. Simmons, "Synthex," *Communications of the ACM*, Vol. 4 , No. 3, p. 140, March 1961. [111]
22. From "Text Retrieval Online: Historical Perspective on Web Search Engines," by Trudi Bellardo Hahn, *ASIS Bulletin*, April/May 1998. Available online at http://www.asis.org/Bulletin/Apr-98/hahn.html. [111]
23. Daniel G. Bobrow, "Natural Language Input for a Computer Problem Solving System," MIT Artificial Intelligence Project Memo 66, Memorandum MAC-M-148, March 30, 1964. Available online at http://dspace.mit.edu/bitstream/handle/1721.1/5922/AIM-066.pdf?sequence=2. An article based on the dissertation is Chapter 3 of Marvin Minsky (ed.), *Semantic Information Processing*, Cambridge, MA: MIT Press, 1968. [111]