ntroduction Conversational agent Dialogue system Machine translation Question answering Dave Bowman: Open the pod bay doors, HAL.The key advantage of probabilistic models is their ability to solve the many kinds of ambiguity problems that we discussed earlier; almost any speech and language processing problem can be recast as: "given N choices for some ambiguous input, choose the most probable one". Finally, vector-space models, based on linear algebra, underlie information retrieval DRAFT 6 Chapter 1. 1.4 Turing test To many, the ability of computers to process language as skillfully as we humans do will signal the arrival of truly intelligent machines. The basis of this belief is the fact that the effective use of language is intertwined with our general cognitive abilities. Among the first to consider the computational implications of this intimate connection was Alan Turing (1950). In this famous paper, Turing introduced what has come to be known as the Turing test. Turing began with the thesis that the question of what it would mean for a machine to think was essentially unanswerable due to the inherent imprecision in the terms machine and think. Instead, he suggested an empirical test, a game, in which a computer's use of language would form the basis for determining if it could think. If the machine could win the game it would be judged intelligent. In Turing's game, there are three participants: two people and a computer. One of the people is a contestant and plays the role of an interrogator. To win, the interrogator must determine which of the other two participants is the machine by asking a series of questions via a teletype. The task of the machine is to fool the interrogator into believing it is a person by responding as a person would to the interrogator's questions. The task of the second human participant is to convince the interrogator that the other participant is the machine, and that she is human. Introduction and many treatments of word meanings. Processing language using any of these models typically involves a search through a space of states representing hypotheses about an input. In speech recognition, we search through a space of phone sequences for the correct word. In parsing, we search through a space of trees for the syntactic parse of an input sentence. In machine trans- lation, we search through a space of translation hypotheses for the correct translation of a sentence into another language. For non-probabilistic tasks, such as tasks involving state machines, we use well-known graph algorithms such as depth-first search. For probabilistic tasks, we use heuristic variants such as best-first and A* search, and rely on dynamic programming algorithms for computational tractability. Machine learning tools such as classifiers and sequence models play a significant role in many language processing tasks. Based on attributes describing each object, a classifer attempts to assign a single object to a single class while a sequence model attempts to jointly classify a sequence of objects into a sequence of classes. For example, in the task of deciding whether a word is spelled correctly or not, clas- sifiers such as decision trees, support vector machines, Gaussian Mixture Models, and logistic regression could be used to make a binary decision (correct or incorrect) for one word at a time. Sequence models such as hidden Markov models, maxi- mum entropy Markov models, and conditional random fields could be used to assign correct/incorrect labels to all the words in a sentence at once. Finally, researchers in language processing use many of the same methodologi- cal tools that are used in machine learning research--the use of distinct training and test sets, statistical techniques like cross-validation, and careful evaluation of trained systems. Language, Thought, and Understanding DRAFT ELIZA Section 1.4. Language, Thought, and Understanding 7 The following interaction from

Turing's paper illustrates the kind of interactions he had in mind. Clearly a convincing simulation of humanity doesn't require being an expert in all domains: Q: Please write me a sonnet on the topic of the Forth Bridge. A: Count me out on this one. I never could write poetry. Q: Add 34957 to 70764. A: (Pause about 30 seconds and then give answer as) 105621. Turing predicted that roughly by the end of the twentieth century a machine with 10 gigabytes of memory would have around a 30% chance of fooling a human interrogator after 5 minutes of questioning. Given the fact that you can fool some of the people all the time, it is not clear how rigorous this particular standard is. Nevertheless, the critical issue for Turing was that using language as humans do is sufficient, by itself, as an operational test for intelligence. Surprisingly, many of the issues raised by Turing became relevant as early as 1966 in the context of the ELIZA program (Weizenbaum, 1966). ELIZA was an early nat- ural language processing system capable of carrying on a limited form of conversation with a user. Consider the following session with a version of ELIZA that imitated the responses of a Rogerian psychotherapist. User1 : ELIZA1: User2 : ELIZA2: User3 : ELIZA3: User4 : ELIZA4: You are like my father in some ways. WHAT RESEMBLANCE DO YOU SEE You are not very aggressive but I think you don't want me to notice that. WHAT MAKES YOU THINK I AM NOT AGGRESSIVE You don't argue with me. WHY DO YOU THINK I DON'T ARGUE WITH YOU You are afraid of me. DOES IT PLEASE YOU TO BELIEVE I AM AFRAID OF YOU As we will discuss in more detail in Ch. 2, ELIZA is a remarkably simple program that makes use of pattern-matching to process the input and translate it into suitable outputs. The success of this simple technique in this domain is due to the fact that ELIZA doesn't actually need to know anything to mimic a Rogerian psychotherapist. As Weizenbaum notes, this is one of the few dialogue genres where the listener can act as if they know nothing of the world. ELIZA's deep relevance to Turing's ideas is that many people who interacted with ELIZA came to believe that it really understood them and their problems. Indeed, Weizenbaum (1976) notes that many of these people continued to believe in ELIZA's abilities even after the program's operation was explained to them. In more recent years, Weizenbaum's informal tests have been repeated in a somewhat more controlled setting. Since 1991, an event known as the Loebner Prize competition has attempted to put various computer programs to the Turing test. Although these contests seem to have little scientific interest, a consistent result over the years has been that even the crudest programs can fool some of the judges some of the time (Shieber, 1994a). Not surpris- ingly, these results have done nothing to quell the ongoing debate over the suitability of the Turing test as a test for intelligence among philosophers and AI researchers (Searle, 1980). DRAFT 8 Chapter 1. Introduction Fortunately, for the purposes of this book, the relevance of these results does not hinge on whether or not computers will ever be intelligent, or understand natural lan- guage. Far more important is recent related research in the social sciences that has confirmed another of Turing's predictions from the same paper. Nevertheless I believe that at the end of the century the use of words and educated opinion will have altered so much that we will be able to speak of machines thinking without expecting to be contradicted. It is now clear that regardless of what people believe or know about the inner workings of computers, they talk about them and interact with them as social entities. People act toward computers as if they were people; they are polite to them, treat them as team members, and expect among other things that computers should be able to understand their needs, and be capable of interacting with them naturally. For example, Reeves and Nass (1996) found that when a computer asked a human to evaluate how well the computer had been doing, the human gives more positive responses than when a differ- ent computer asks the same questions. People seemed to be afraid of being impolite. In a different experiment, Reeves and Nass found that people also give computers higher performance ratings if the computer has recently said something flattering to the hu- man. Given these predispositions, speech and language-based systems may provide many users with the most natural interface for many applications. This fact has led to a long-term focus in the field on the design of conversational agents, artificial entities that communicate conversationally. 1.5 The State of the Art We can only see a short distance ahead, but we can see plenty there that needs to be done. Alan Turing. This is an exciting time for the field of speech and language processing. The startling increase in computing resources available to the average computer user, the rise of the Web as a massive source of information and the increasing availability of wireless mobile access have all placed speech and language processing applications in the technology spotlight. The following are examples of some currently deployed systems that reflect this trend: o Travelers calling Amtrak, United Airlines and other travel-providers interact with conversational agents that guide them through the process of making reser- vations and getting arrival and departure information. o Luxury car makers such as Mercedes-Benz models provide automatic speech recognition and text-to-speech systems that allow drivers to control their envi- ronmental, entertainment and navigational systems by voice. A similar spoken dialogue system has been deployed by astronauts on the International Space Sta- tion . o Blinkx and other video search companies provide search services for million of hours of video on the Web by using speech recognition technology to capture the words in the sound track. DRAFT Section 1.6. Some Brief History 9 o Google provides cross-language information retrieval and translation services where a user can supply queries in their native language to search collections in another language. Google translates the query, finds the most relevant pages and then automatically translates them back to the user's native language. o Large educational publishers such as Pearson, as well as testing services like ETS, use automated systems to analyze thousands of student essays, grading and assessing them in a manner that is indistinguishable from human graders. o Interactive tutors, based on lifelike animated characters, serve as tutors for chil- dren learning to read (Wise et al., 2007). o Text analysis companies such as Nielsen Buzzmetrics, Umbria, and Collective Intellect provide marketing intelligence based on automated measurements of user opinions, preferences, attitudes as expressed in weblogs, discussion forums and user groups. 1.6 Some Brief History Historically, speech and language processing has been treated very differently in com- puter science, electrical engineering, linguistics, and psychology/cognitive science. Because of this diversity, speech and language processing encompasses a number of different but overlapping fields in these different departments: computational linguis- tics in linguistics, natural language processing in computer science, speech recogni- tion in electrical engineering, computational psycholinguistics in psychology. This section summarizes the different historical threads which have given rise to the field of speech and language processing. This section will provide only a sketch, but many of the topics listed here will be covered in more detail in subsequent chapters. 1.6.1 Foundational Insights: 1940s and 1950s The earliest roots of the field date to the intellectually fertile period just after

World War II that gave rise to the computer itself. This period from the 1940s through the end of the 1950s saw intense work on two foundational paradigms: the automaton and probabilistic or information-theoretic models. The automaton arose in the 1950s out of Turing's (1936) model of algorithmic computation, considered by many to be the foundation of modern computer science. Turing's work led first to the McCulloch-Pitts neuron (McCulloch and Pitts, 1943), a simplified model of the neuron as a kind of computing element that could be described in terms of propositional logic, and then to the work of Kleene (1951) and (1956) on finite automata and regular expressions. Shannon (1948) applied probabilistic models of discrete Markov processes to automata for language. Drawing on the idea of a finite-state Markov process from Shannon's work, Chomsky (1956) first considered finite-state machines as a way to characterize a grammar, and defined a finite-state language as a language generated by a finite-state grammar. These early models led to the field of formal language theory, which used algebra and set theory to define formal languages as sequences of symbols. This includes the context-free grammar, DRAFT 10 Chapter 1. Introduction first defined by Chomsky (1956) for natural languages but independently discovered by Backus (1959) and Naur et al. (1960) in their descriptions of the ALGOL programming language. The second foundational insight of this period was the development of probabilistic algorithms for speech and language processing, which dates to Shannon's other con- tribution: the metaphor of the noisy channel and decoding for the transmission of language through media like communication channels and speech acoustics. Shannon also borrowed the concept of entropy from thermodynamics as a way of measuring the information capacity of a channel, or the information content of a language, and performed the first measure of the entropy of English using probabilistic techniques. It was also during this early period that the sound spectrograph was developed (Koenig et al., 1946), and foundational research was done in instrumental phonetics that laid the groundwork for later work in speech recognition. This led to the first machine speech recognizers in the early 1950s. In 1952, researchers at Bell Labs built a statistical system that could recognize any of the 10 digits from a single speaker (Davis et al., 1952). The system had 10 speaker-dependent stored patterns roughly representing the first two vowel formants in the digits. They achieved 97-99% accuracy by choosing the pattern that had the highest relative correlation coefficient with the input. 1.6.2 The Two Camps: 1957-1970 By the end of the 1950s and the early 1960s, speech and language processing had split very cleanly into two paradigms: symbolic and stochastic. The symbolic paradigm took off from two lines of research. The first was the work of Chomsky and others on formal language theory and generative syntax throughout the late 1950s and early to mid 1960s, and the work of many linguistics and computer sci- entists on parsing algorithms, initially top-down and bottom-up and then via dynamic programming. One of the earliest complete parsing systems was Zelig Harris's Trans- formations and Discourse Analysis Project (TDAP), which was implemented between June 1958 and July 1959 at the University of Pennsylvania (Harris, 1962).2 The sec- ond line of research was the new field of artificial intelligence. In the summer of 1956 John McCarthy, Marvin Minsky, Claude Shannon, and Nathaniel Rochester brought together a group of researchers for a two-month workshop on what they decided to call artificial intelligence (AI). Although AI always included a minority of researchers focusing on stochastic and statistical algorithms (including probabilistic models and neural nets), the major focus of the new field was the work on reasoning and logic typified by Newell and Simon's work on the Logic Theorist and the General Problem Solver. At this point early natural language understanding systems were built. These simple systems worked in single domains mainly by a combination of pattern matching and keyword search with simple heuristics for reasoning and question-answering. By the late 1960s more formal logical systems were developed. The stochastic paradigm took hold mainly in departments of statistics and of elec- 2 This system was reimplemented recently and is described by Joshi and Hopely (1999) and Karttunen (1999), who note that the parser was essentially implemented as a cascade of finite-state transducers. DRAFT Section 1.6. Some Brief History 11 trical engineering. By the late 1950s the Bayesian method was beginning to be applied to the problem of optical character recognition. Bledsoe and Browning (1959) built a Bayesian system for text-recognition that used a large dictionary and computed the likelihood of each observed letter sequence given each word in the dictionary by mul- tiplying the likelihoods for each letter. Mosteller and Wallace (1964) applied Bayesian methods to the problem of authorship attribution on The Federalist papers. The 1960s also saw the rise of the first serious testable psychological models of human language processing based on transformational grammar, as well as the first on-line corpora: the Brown corpus of American English, a 1 million word collection of samples from 500 written texts from different genres (newspaper, novels, non-fiction, academic, etc.), which was assembled at Brown University in 1963-64 (Kuc?era and Francis, 1967; Francis, 1979; Francis and Kuc?era, 1982), and William S. Y. Wang's 1967 DOC (Dictionary on Computer), an on-line Chinese dialect dictionary. 1.6.3 Four Paradigms: 1970-1983 The next period saw an explosion in research in speech and language processing and the development of a number of research paradigms that still dominate the field. The stochastic paradigm played a huge role in the development of speech recog- nition algorithms in this period, particularly the use of the Hidden Markov Model and the metaphors of the noisy channel and decoding, developed independently by Jelinek, Bahl, Mercer, and colleagues at IBM's Thomas J. Watson Research Center, and by Baker at Carnegie Mellon University, who was influenced by the work of Baum and colleagues at the Institute for Defense Analyses in Princeton. AT&T's Bell Laborato- ries was also a center for work on speech recognition and synthesis; see Rabiner and Juang (1993) for descriptions of the wide range of this work. The logic-based paradigm was begun by the work of Colmerauer and his col- leagues on Q-systems and metamorphosis grammars (Colmerauer, 1970, 1975), the forerunners of Prolog, and Definite Clause Grammars (Pereira and Warren, 1980). In- dependently, Kay's (1979) work on functional grammar, and shortly later, Bresnan and Kaplan's (1982) work on LFG, established the importance of feature structure unifica- tion. The natural language understanding field took off during this period, beginning with Terry Winograd's SHRDLU system, which simulated a robot embedded in a world of toy blocks (Winograd, 1972a). The program was able to accept natural language text commands (Move the red block on top of the smaller green one) of a hitherto unseen complexity and sophistication. His system was also the first to attempt to build an extensive (for the time) grammar of English, based on Halliday's systemic grammar. Winograd's model made it clear that the problem of parsing was well enough under- stood to begin to focus on semantics and discourse models. Roger Schank and his colleagues and students (in what was often referred to as the Yale School) built a se- ries of language understanding programs that focused on human conceptual knowledge such as scripts, plans and goals, and human memory organization (Schank and Albel- son, 1977; Schank and Riesbeck, 1981; Cullingford, 1981; Wilensky, 1983; Lehnert, 1977). This work often used network-based semantics (Quillian, 1968; Norman and Rumelhart, 1975; Schank, 1972; Wilks, 1975c, 1975b; Kintsch, 1974) and began to DRAFT 12 Chapter 1. Introduction incorporate Fillmore's notion of case roles (Fillmore, 1968) into their representations (Simmons, 1973). The logic-based and natural-language understanding paradigms were unified in systems that used predicate logic as a semantic representation, such as the LUNAR question-answering system (Woods, 1967, 1973). The discourse modeling paradigm focused on four key areas in discourse. Grosz and her colleagues introduced the study of substructure in discourse, and of discourse focus (Grosz, 1977a; Sidner, 1983), a number of researchers began to work on au- tomatic reference resolution (Hobbs, 1978), and the BDI (Belief-Desire-Intention) framework for logic-based work on speech acts was developed (Perrault and Allen, 1980; Cohen and Perrault, 1979). 1.6.4 Empiricism and Finite State Models Redux: 1983-1993 This next decade saw the return of two classes of models which had lost popularity in the late 1950s and early 1960s, partially due to theoretical arguments against them such as Chomsky's influential review of Skinner's Verbal Behavior (Chomsky, 1959b). The first class was finite-state models, which began to receive attention again after work on finite-state phonology and morphology by Kaplan and Kay (1981) and finite-state models of syntax by Church (1980). A large body of work on finite-state models will be described throughout the book. The second trend in this period was what has been called the "return of empiri- cism"; most notably here was the rise of probabilistic models throughout speech and language processing, influenced strongly by the work at the IBM Thomas J. Watson Research Center on probabilistic models of speech recognition.Just a few of the "multiples" to be discussed in this book include the application of dynamic programming to sequence comparison by Viterbi, Vintsyuk, Needleman and Wunsch, Sakoe and Chiba, Sankoff, Reichert et al., and Wagner and Fischer (Chapters 3, 5 and 6) the HMM/noisy channel model of speech recognition by Baker and by Jelinek, Bahl, and Mercer (Chapters 6, 9, and 10); the development of context-free grammars by Chomsky and by Backus and Naur (Chapter 12); the proof that Swiss-German has a non-context-free syntax by Huybregts and by Shieber (Chapter 15); the application of unification to language processing by DRAFT 14 Chapter 1. Introduction Colmerauer et al. and by Kay in (Chapter 16). Are these multiples to be considered astonishing coincidences? A well-known hy- pothesis by sociologist of science Robert K. Merton (1961) argues, quite the contrary, that all scientific discoveries are in principle multiples, including those that on the surface appear to be singletons. Of course there are many well-known cases of multiple discovery or invention; just a few examples from an extensive list in Ogburn and Thomas (1922) include the multiple invention of the calculus by Leibnitz and by Newton, the multiple development of the theory of natural selection by Wallace and by Darwin, and the multiple invention of the telephone by Gray and Bell.3 But Merton gives a further array of evidence for the hypothesis that multiple discovery is the rule rather than the exception, including many cases of putative singletons that turn out be a rediscovery of previously unpublished or perhaps inaccessible work. An even stronger piece of evidence is his ethnomethodolog- ical point that scientists themselves act under the assumption that multiple invention is the norm. Thus many aspects of scientific life are designed to help scientists avoid be- ing "scooped"; submission dates on journal articles; careful dates in research records; circulation of preliminary or technical reports.Importantly, included among these materials were annotated collections such as the Penn Treebank (Marcus et al., 1993), Prague Dependency Treebank (Hajic?, 1998), PropBank (Palmer et al., 2005), Penn Discourse Treebank (Miltsakaki et al., 2004b), RSTBank (Carlson et al., 2001) and TimeBank (Pustejovsky et al., 2003b), all of which layered standard text sources with various forms of syntactic, semantic and pragmatic annotations.These resources also promoted the establishment of additional competitive evaluations for parsing (Dejean and Tjong Kim Sang, 2001), information extraction (NIST, 2007a; Sang, 2002; Sang and De Meulder, 2003), word sense disambiguation (Palmer et al., 2001a; Kilgarriff and Palmer, 2000), question an- swering (Voorhees and .Tice, 1999), and summarization Dang (2006).DRAFT Section 1.6