Chapter 00

# The Anatomy of A.L.I.C.E.

Dr. Richard S. Wallace
*A.L.I.C.E. Artificial Intelligence Foundation, Inc.*

Key words: Artificial intelligence, natural language, chatterbot, chat robot, softbot, bot, Artificial Intelligence Markup Language (AIML), Markup Languages, XML, HTML, philosophy of mind, consciousness, dualism, behaviorism, functionalism, reductionism, recursion, stimulus-response, pattern recognition, machine intelligence, Turing Test, Loebner Prize, free software, open source, A.L.I.C.E., Artificial Linguistic Internet Computer Entity, talking computer, deception, targeting

Abstract: This paper is a technical presentation of Artificial Linguistic Internet Computer Entity (A.L.I.C.E.) and Artificial Intelligence Markup Language (AIML), set in context by historical and philosophical ruminations on human consciousness. A.L.I.C.E., the first AIML-based personality program, won the Loebner Prize as "the most human computer" at the annual Turing Test contests in 2000 and 2001. (Loebner 2002) The program, and the organization that develops it, is a product of the world of free software. More than 500 volunteers from around the world have contributed to her development. This paper describes the history of A.L.I.C.E. and AIML-free software since 1995, noting that the theme and strategy of deception and pretense upon which AIML is based can be traced through the history of artificial intelligence research. This paper goes on to show how to use AIML to create robot personalities like A.L.I.C.E. that pretend to be intelligent and self-aware. The bot 'personality' is a set of AIML files consisting of simple stimulus-response modules called categories. Each <category> contains a <pattern>, or "stimulus," and a <template>, or "response." AIML software stores the stimulus-response categories in a tree managed by an object called the Graphmaster. When a bot client inputs text as a stimulus, the Graphmaster searches the categories for a matching <pattern>, along with any associated context, and then outputs the associated <template> as a response. These categories can be structured to produce more complex humanlike responses with the use of a very few markup tags. AIML bots make extensive use of the

multi-purpose recursive <srai> tag, as well as two AIML context tags, <that> and <topic>. Conditional branching in AIML is implemented with the <condition> tag. AIML implements the ELIZA personal pronoun swapping method with the <person> tag. Bot personalities are created and shaped through a cyclical process of supervised learning called Targeting. Targeting is a cycle incorporating client, bot, and botmaster, wherein client inputs that find no complete match among the categories are logged by the bot and delivered as Targets the botmaster, who then creates suitable responses, starting with the most common queries. The Targeting cycle produces a progressively more refined bot personality. The art of AIML writing is most apparent in creating default categories, which provide noncommittal replies to a wide range of inputs. The paper winds up with a survey of some of the philosophical literature on the question of consciousness. We consider Searle's Chinese Room, and the view that natural language understanding by a computer is impossible. We note that the proposition "consciousness is an illusion" may be undermined by the paradoxes it apparently implies. We conclude that A.L.I.C.E. does pass the Turing Test, at least, to paraphrase Abraham Lincoln, for some of the people some of the time.

TABLE OF CONTENTS

# 1. INTRODUCTION

A.L.I.C.E. is an artificial intelligence natural language chat robot based on an experiment specified by Alan M. Turing in 1950. The A.L.I.C.E. software utilizes AIML, an XML language we designed for creating stimulus-response chat robots.

Some view A.L.I.C.E. and AIML as a simple extension of the old ELIZA psychiatrist program. The comparison is fair regarding the stimulus-response architecture. But the A.L.I.C.E. bot has at present more than 40,000 categories of knowledge, whereas the original ELIZA had only about 200. Another innovation was provided by the web, which enabled natural language sample data collection possible on an unprecedented scale.

A.L.I.C.E. won the Loebner Prize, an annual Turing Test, in 2000 and 2001. Although no computer has ever ranked higher than the humans in the contest she was ranked “most human computer” by the two panels of judges. What it means to “Pass the Turing Test” is not so obvious. Factors such as the age, intellect and expectations of the judges have tremendous impact on their perceptions of intelligence. Alan Turing himself did not describe only one “Turing Test.” His original imitation game involved determining the gender of the players, not their relative humanness.

The model of learning in A.L.I.C.E. is called supervised learning because a person, the botmaster, plays a crucial role. The botmaster monitors the robot’s conversations and creates new AIML content to make the responses more appropriate, accurate, believable, or “human,” or whatever the botmaster intends. We have developed algorithms for automatic detection of patterns in the dialog data. This process, called “Targeting,” provides the botmaster with new input patterns that do not already have specific replies, permitting a process of almost continuous supervised refinement of the bot.

Some have argued that Turing, when he predicted that a machine could play his game in “50 years” after his 1950 paper, envisioned something more like a general purpose learning machine, which does not yet exist. The concept is simple enough: build a robot to grow like a child, able to be taught language the way we are. In our terms, the role of the botmaster would be fully automated. But even a child does not, or at least should not, go forth into the world, unprotected, to learn language “on the street,” without supervision.

Automatic generation of chat robot questions *and* answers appears likely to raise the same trust issues forced upon the abandoned child. People are

simply too untrustworthy in the "facts" that they would teach the learning machine. Many clients try to deliberately sabotage the bot with false information. There would still have to be an editor, a supervisor, a botmaster or teacher to cull the wheat from the chaff.

The brain of A.L.I.C.E. consists of roughly 41,000 elements called categories. Each category combines a question and answer, or stimulus and response, called the "pattern" and "template" respectively. The AIML software stores the patterns in a tree structure managed by an object called the Graphmaster, implementing a pattern storage and matching algorithm. The Graphmaster is compact in memory, and permits efficient pattern matching time.

## 2. THE PROBLEM

Susan Sterrett's careful reading of Turing's 1950 paper reveals a significant distinction between two different versions of what has come to be known as the Turing Test (Sterrett 2000). The first version, dubbed the Original Imitation Game (OIG), appears on the very first page of *Computing Machinery and Intelligence* (Turing 1950). The OIG has three players: a man (A), a woman (B), and a third person (C) of either sex. The third player (C) is called the interrogator, and his function is to communicate with the other two, through what would nowadays be called a text-only instant messaging chat interface, using two terminals (or today perhaps, two windows) labeled (X) and (Y). The interrogator must decide whether (X) is (A) and (Y) is (B), or (X) is (B) and (Y) is (A), in other words which is the man and which is the woman. The interrogator's task is complicated by the man (A), who Turing says should reply to the interrogator with lies and deceptions. For example, if the man is asked, "are you a man or a woman?," he might reply, "I am a woman."

Putting aside the gender and social issues raised by the OIG, consider the OIG as an actual scientific experiment. Turing's point is that if we were to actually conduct the OIG with a sufficiently large sample of subjects playing the parts of (A), (B), and (C), then we could measure a specific percentage M of the time that, on average, the interrogator misidentifies the woman, so that 100-M% of the time she is identified correctly. Given enough trials of the OIG, at least in a given historical and cultural context, the number M ought to be a fairly repeatable measurement.

Now, as Turing said, consider replacing the man (A) with a computer. What would happen if we tried the experiment with a very simple minded program like ELIZA? In that case, the interrogator (C) would identify the woman correctly (nearly) 100 percent of the time, so that M=0. The ELIZA

program would not do well in the OIG, but as the variety and quality of machine's responses begin to approach those of the lying man, the measured percentage of incorrect identification ought to be closer and closer to the M measured with the man playing (A).

Much later in the 1950 paper, in section 5, Turing describes a second game more like the concept of a "Turing Test" as most engineering schools teach it. The setup is similar to the OIG, but now gender plays no role. The player (B) is called "a man" and the player (A) is always a computer. The interrogator must still decide whether (X) is (A) and (Y) is (B), or (X) is (B) and (Y) is (A), in other words which is the man and which is the machine. Sterrett calls this second game the Standard Turing Test (STT).

Whole academic conferences have been devoted to answering the question of what Turing meant by the Turing Test. In a radio interview taped by the BBC, Turing describes a game more like the STT, but in the paper he gives more prominence to the OIG. Unlike the OIG, the STT is not a good scientific experiment. What does it mean to "pass" the STT? Must the interrogator identify the machine correctly 50% of the time, or 100%? For how long must the machine deceive the interrogator? Finally, does the interrogator know in advance that he is trying to "out"(Zdenek 2000) the robot, or that one of the players is a machine at all?

Unfortunately the STT, though flawed as an experiment, has come to be popularized as the modern "Turing Test." The STT is the basis of real-world Turing Tests including the Loebner Prize, won by A.L.I.C.E. in 2000 and 2001. Although she performs well in STT style contests, the A.L.I.C.E. personality is actually designed to play the OIG. She is a machine, pretending to be a man, pretending to be a woman. Her technology is based on the simplest A.I. program of all, the old ELIZA psychiatrist.

## 3. THE PSYCHIATRIST

The story of Joseph Weizenbaum is in many ways almost as interesting as that of Alan Turing, although not so tragic. One almost pictures Weizenbaum as a latter-day Luddite, smashing artificial intelligence the way British tradesmen smashed factories in reaction to the Industrial Revolution. But unlike the Luddites, Weizenbaum himself was the inventor of the machine. It would be more akin to the Wright Brothers destroying their prototype airplane after its first flight, saying "this technology is too dangerous. We must put the genie back in the bottle."

A typical conversation with ELIZA is the following:

C: Well, my boyfriend made me come here.

R: Your boyfriend made you come here.
C: I am unhappy.
R: Do you think coming here will help you not to be unhappy?
C: I need some help, that much seems certain.
R: What would it mean to you if you got some help?
C: Perhaps I could learn to get along with my mother.
R: Tell me more about your family.
C: You are like my father in some ways.
R: What resemblance do you see?
C: My father is afraid of everybody.
R: What else comes to mind when you think of your father.
C: Bullies.
R: Does that have anything to with the fact that your boyfriend made you come here?

Note: In all sample dialogues following, the "R:" stands for "Robot:" and the "C:" stands for "Client."

One of ELIZA's fundamental "tricks" is the pronoun reversal exhibited in the first exchange. Mapping "my" to "your" and "me" to "you," the robot echoes the client input by exchanging first and second personal pronouns. Other replies are generic defaults, giving the illusion of some understanding. If the client says anything with the prefix "I need some X," the robot replies with "What would it mean to you if you got some X?" It doesn't matter whether X is help, money, food, water, love or time. The same answer will cover almost all the likely inputs.

Still other ELIZA replies are based on simple keyword recognition, as in the exchange about the client's mother, when the robot says, "Tell me more about your family." The appearance of the keyword "mother" anywhere in the input may have triggered this response. ELIZA has a limited memory of the conversation state, as well. When confronted with the unrecognized input "Bullies," she responds by raising the previously stored topic.

As unlikely as it sounds today, Weizebaum pulled the plug on ELIZA (Weizenbaum 1976). He was horrified that anyone would actually believe this simple program said anything about intelligence, let alone had any. Weizenbaum tells us that he was shocked by the experience of releasing ELIZA, also known as "Doctor," for use by nontechnical staff at MIT. Secretaries and nontechnical staff thought the machine was a "real" therapist, and spent hours revealing their personal problems to the program. When Weizenbaum informed a secretary that, of course, he had access the logs of all the conversations, she reacted with outrage at this invasion of privacy. Weizenbaum was shocked that such a simple program could deceive a naive client into revealing personal information.

What Weizenbaum found especially revolting was that the Doctor's patients believed the robot really understood their problems. Even some psychiatrists seriously believed the robot therapist could help patients in a constructive way. Weizenbaum's reaction might be best understood like that of a Western physician's disapproval of herbal medicine, or an astronomer's disdain for astrology.

The back cover of the paper edition of Weizenbaum's Computer Power and Human Reason (Weizenbaum, 1976) gives us a feeling for the general attitude toward the book at the time of its release:

> "Dare I say it? This is the best book I have read on the impact of computers on society, and on technology, and man's image of himself."—Keith Oakley, *Psychology Today*
>
> "A thoughtful blend of insight, experience, anecdote, and passion that will stand for a long time as the definitive integration of technological and human thought."—American Mathematical Monthly
>
> "Superb ... the work of a man who is struggling with the utmost seriousness to save our humanity from the reductionist onslaught of one of the most prestigious, and richly funded technologies of our time."—Theodore Piszak, The Nation.

*Computer Power and Human Reason* seems a bit quaint today, much the same as Turing's 1950 paper does. For one thing, Weizenbaum perceived his mission as partly to educate an uninformed public about computers. Presumably the uninformed public was confusing science fiction with reality. Most of the book is devoted to explaining how a computer works, perhaps a necessary primer in 1976, but today more appropriate in a book called something like, "Computers for Dummies."

Two chapters of *Computer Power and Human Reason* are devoted to a humanist attack on artificial intelligence, on ELIZA specifically, and on computer science research in general.

Most contemporary scholars did not need much convincing that ELIZA was at best a gimmick, at worst a hoax, and in any case not a "serious" artificial intelligence project. Yet ELIZA is perhaps the most widely distributed program in the history of artificial intelligence. The irony of Joseph Weizenbaum is that by failing to promote his own technology, indeed by encouraging his own critics, he successfully blocked much further investigation into what would prove to be one of the most persistently interesting demonstrations to emerge from the MIT AI Lab.

## 4. POLITICIANS

Client: Do you think President Clinton should be impeached?

Robot: It depends what you mean by “thinking.”

Politicians, at least those in our society, never seem to give a straight answer to a question. If a journalist asks a specific question, the politician answers with a “sound bite” or short, memorized speech which is related to, but does not necessarily answer, the reporter’s question. If asked about schools for example, the politician may reply with a memorized policy statement on education policy. It is as though the response was merely triggered by the keyword “school” in the question. No creative or spontaneous thinking is required.

One aspect of the art of writing AIML is composing brief, concise, interesting, grammatically correct and sometimes humorous default responses, which work in for a wide variety of inputs matching a single pattern. President Clinton inadvertently revealed this type of automatic reply when he uttered the famous quote, “It depends on what the meaning of ‘is’ is.” This could be a default response to any input beginning with the word “is.” Like the politician’s replies, the default responses should indicate an understanding of the question, but not offend anyone.

There is a democratic aspect to A.L.I.C.E. Born out of frustration about public apathy toward the existing attempts at artificial intelligence, A.L.I.C.E. is designed to answer the question, what do people expect an A.I. ought to be able to say? It turns out that there is a Zipf like distribution of the queries people make to the robot. The appearance of the Zipf distribution leads to a natural strategy for working on the problem: write replies for the most common queries first.

Despite the obvious simplicity of this approach, largely borrowed from the ELIZA psychiatrist program, the prevailing prejudice for many years, at least among the academic elite, has derided ELIZA as a “toy,” too simple to form the basis of a practical natural language understanding system. One school of thought advocated an approach based on limited domains of discourse, in which practical systems might converse about one subject like moon rocks, blocks, or Star Trek. This school would have us lower our expectations for general natural language understanding however. Another school favors development of large scale ontologies of “commonsense knowledge,” thought to be necessary for understanding everyday language. Still a third school advocates building something like the general learning machine, which can be taught language like a child. It goes without saying

that no other school has produced any machine capable of anything close to the performance of A.L.I.C.E. in Turing's Imitation Game.

## 5. PARTIES

Social situations prove awkward for many of us. The author, challenged by an almost total apathy toward organized sports of any kind, navigated the minefield of adolescence in America by making use of a technique he called "sports talk." When cornered with the question, "Did you see the game last night?," the author would always reply, "No, who won?" Many sports questions, such as "Who do you want to win the super bowl?" can be deflected with dry humor like, "the team that scores the most points." When confronted with a question about a specific sports player, like "What do you think of Joe Montana this season?," can be handled believably with imaginary biographical data, as in, "I heard his brother was a boxer." By remembering the previous utterance, the technique may even be extended to more involved dialogues:

C: You should have seen the game last night.
R: What game?
C: Basketball.
R: Sorry I missed it.

Conversations with strangers may be initiated with appropriately selected "pickup lines." Early in the history of A.L.I.C.E. the bot also initiated conversations by randomly selecting one of these lines. It was then possible to measure the average dialogue length of conversations initiated with each line. Although the original data was sadly lost, the experiment may be easily repeated with the latest generation of AIML software. In our brief test, the pickup line with the longest average dialogue length was, "What is your favorite movie?" Among about 40 pickup lines tested, the lowest ranked was, "What do you think caused World War I?" Another poor choice was, "Are you a man or a woman?," which is ironic because this may be the single most often used pickup line in human online chat ("r u m/f?").

Experience with A.L.I.C.E. indicates that most casual conversation is "stateless," that is, each reply depends only on the current query, without any knowledge of the history of the conversation required to formulate the reply. Indeed in human conversation it often seems that we have the reply "on the tip of the tongue" even before the interlocutor has completed his query. Occasionally following the dialogue requires a conversational memory of one more level, implemented in AIML with <that>. When asking a question,

the question must be remembered long enough to be combined with the answer. These same remarks are not necessarily true in situations requiring highly structured dialogue, such as courtrooms or classrooms. But in the informal party situation human conversation does not appear to go beyond simple stimulus-response, at least not very often.

How often do we find ourselves half-listening, half-thinking about something else, in the midst of a conversation? Offering the occasional nod, "uh-huh," or other gesture indicating understanding, we put on the illusion of understanding and "being human." Questions are often repeated in informal human chat as well.

C: Where did you go to college?
R: I told you before, it was Cambridge.
C: Oh, that's right. I'm sorry.

With her unstructured approach to conversations, A.L.I.C.E. is also capable of the kind of passive-aggressive data collection characteristic of human conversations. A totally passive data collection device is like a web guestbook, where there are no constraints placed on the data collected. The client may write anything in a guestbook. An example of an aggressive data collection device is a nitpicky form, which may not even be submitted until every field is filled.

Humans and A.L.I.C.E. can collect a lot of personal information through the use of leading questions in the conversation, such as "How old are you?" or "Are you a student?" We call this type of data collection, passive-aggressive, because it combines elements of the passive guestbook with those of the aggressive form. Provided that bot chats with enough clients, the passive-aggressive method can collect a statistically significant amount of client data. Using this type of data collection we have been able to ascertain that about half the clients of A.L.I.C.E. are under 18, for example.

## 6. THE PROFESSOR

Every experienced professor knows that there is a Zipf distribution of questions asked by students in class. The single most common question is universally, "Will this be on the test?" The lecturer's job is like that of a FAQ bot or politician, to memorize the answers to all of the most commonly asked questions, and even to match an ambiguous question with one he already knows the answer to. In the rare event that the student confronts the teacher with a question he cannot answer, the professor supplies a default response indicating that he understood the question and may provide an

answer at a later time. One good default response like that is, “That is not my area of expertise.”

A general downturn in artificial intelligence and robotics roughly coincided with the end of the Cold War, as governments and corporations reduced the amount of funding available for this technology. The “richly funded” field of 1976 became more like a Darwinian struggle for diminishing resources. One positive outcome was the brief heyday of “robot minimalism,” a design philosophy based on low-cost parts, commodity computers, low-bandwidth sensing, and general simplicity in design and engineering. It was a moment when Occam’s razor could cut away much of the needless complexity that had accumulated over the previous decades. Although robot minimalism subsequently fell out of favor, it became a significant influence on the development of A.L.I.C.E.

We used to say there was *no* theory behind A.L.I.C.E., no neural networks, no knowledge representation, no deep search, no genetic algorithms and no parsing. Then we discovered that there was a theory circulating in applied A.I., called Case-Based Reasoning (CBR) [CBR?? reference] that closely resembled the stimulus-response structure of A.L.I.C.E. The CBR cases correspond to the AIML categories.

## 7. PNAMBIC

> “PNAMBIC—(acronym) Pay No Attention to that Man Behind the Curtain [from The Wizard of Oz]. Denoting any supposedly fully automated system that in fact requires human intervention to achieve the desired result.”—New Hacker’s Dictionary

A.L.I.C.E. was not the original name of A.L.I.C.E. The first prototype was called PNAMBIC, in tribute to the hoaxes, deceptions and tricks that have littered the history of artificial intelligence. But the machine hosting PNAMBIC was already named Alice by a forgotten systems administrator, so people began to call her “Alice.” At that point, we invented the “retronym”: Artificial Linguistic Internet Computer Entity. Yet A.L.I.C.E. is possibly the first A.I. technology to embrace this tradition of deception openly.

The tradition goes back to Baron von Kempelen and his 18th century “Chess Playing Automaton.” [add reference??] Also known as the “Strange Turk,” this device appeared to play decent games of chess against any human challenger. Kemepelen utilized a standard magician’s trick, opening first one cabinet door and then closing it, and opening another one, to reveal the “mechanism” inside. According to one legend, the empress of Russia

ordered the machine shot, killing the hapless vertically challenged Polish operator hidden inside.

A book of fiction and poetry, supposedly written by an A.I. named RACTER, caused a minor sensation upon its release in 1984. Later proved to be a hoax (Barger 1993), the book (Chamberlain 1978), called "The Policeman's Beard is Half Constructed," by William Chamberlain, nevertheless speaks to the public's willingness to suspend its disbelief about artificial intelligence. Who can blame them? Hollywood, more than anyone, has done the most to raise the public expectations for A.I. and robots.

The following example illustrates the flavor of the stories told by RACTER. "Bill sings to Sarah, Sarah sings to Bill. Perhaps they will do other dangerous things together. They may eat lamb or stroke each other. They may chant of their difficulties and their happiness. They have love but they also have typewriters. That is interesting." RACTER was a PNAMBIC because obtaining these results required considerable human intervention. At the very least, a human editor reviewed many random examples, looking for sensible ones like the story above.

According to one A.I. urban legend, apparently not documented elsewhere, a famous natural language researcher was embarrassed around the same time, when it became apparent to his audience of Texas bankers that the robot was consistently responding to the *next* question he was about to ask. He was demonstrating a PNAMBIC, a demonstration of natural language understanding that was in reality nothing but a simple script.

The very existence of PNAMBIC as a meme suggests a widespread understanding of how deception might play a role in automated systems. In the rush to complete work and produce demos before bureaucratic deadlines, it is tempting to cut corners. Such deceptions may even be rationalized if they seem justified as inessential to the experimental outcome.

The PNAMBIC meme begs the question, just how much of the published research in the history of artificial intelligence ought not to be regarded as a swindle? In certain academic circles, playing a political charade has replaced actual scientific research as a career objective. The games people play to secure funding, be published in academic journals, be promoted in the academic world; "the old boy's network" and predominance of political correctness, make much of the body of today's publicly funded research highly suspect.

It was against this backdrop that the first real world Turing Test, the Loebner Contest, was held in Boston in 1991. None of the competing programs came close to the performance of the human confederates, but the one ranked highest was based on the simple ELIZA psychiatrist program. The same programmer in fact won the bronze medal in each of the first four annual contests.

## 8. THE PRIZE

Hugh Loebner is an independently wealthy, eccentric businessman, activist and philanthropist. In 1990 Dr. Loebner, who holds a Ph.D. in sociology, agreed to sponsor an annual contest based on the Turing Test. The contest awards medals and cash prizes for the "most human" computer.

Since its inception, the Loebner contest has been a magnet for controversy. One of the central disputes arose over Hugh Loebner's decision to award the Gold Medal and $100,000 top cash prize only when a robot is capable of passing an "audio-visual" Turing Test. The rules for this Grand Prize contest have not even been written yet. So it remains unlikely that anyone will be awarded the gold Loebner medal in the near future.

The Silver and Bronze medal competitions are based on the STT. In 2001, eight programs played alongside two human confederates. A group of 10 judges rotated through each of ten terminals and chatted about 15 minutes with each. The judges then ranked the terminals on a scale of "least human" to "most human." Winning the Silver Medal and its $25,000 prize requires that the judges rank the program higher than half the human confederates. In fact one judge ranked A.L.I.C.E. higher than one of the human confederates in 2001. Had all the judges done so, she might have been eligible for the Silver Medal as well, because there were only two confederates.

## 9. THE PORTAL

When the World Wide Web appeared in 1994, our initial reaction was to adopt a series of micro-robot experiments then underway in our lab for the web. These culminated in Labcam, an online pan-tilt camera with remote actuator control. Clients could select a point on the image with a mouse, and the camera would move to make that point the center of the next view. This awakened our interest in statistical analysis of web client behavior.

Around the same time, many observers began to notice that, on a given web site, there tends to be an uneven distribution of document accesses. If the documents are ranked by access count, and the number of accesses plotted as a bar graph, the distribution resembles the curve y=1/x. If the curve is plotted in log-log coordinates, it appears as a straight line with a slope of -1.

What we were seeing was an example of Zipf's Law (Zipfref??). According to Zipf, this curve is characteristic of natural languages. If a language is purely random, with each symbol or word having equal probability, the curve would be a flat, horizontal line. Because Zipf's Law is

found to apply to a variety of natural phenomena, it is not entirely surprising that it should be observed in the pattern of web document access.

The Web created for the first time the ability to conduct an artificial intelligence experiment along with thousands, even millions, of clients repeatedly testing the system. Previous chat bots had used other IP protocols such as telnet (Mauldin 1996) to reach large audiences, but the Web created the opportunity to collect natural language samples on an unprecedented scale.

If there was any significant innovation after ELIZA, it was this. There is a world of difference between writing 10,000 questions and answers for a bot, versus knowing in advance what the top 10,000 most likely questions will be. A.L.I.C.E. replies were developed directly in response to what people say.

The Internet created another opportunity as well. It became possible to recruit hundreds of volunteer developers worldwide, to work together in a totally new type of research organization.

## 10. PENGUINS

The story of A.L.I.C.E. and AIML cannot be complete without a visit to the world of free software and open source. Because the AIML standard and software was developed by a worldwide community of volunteers, we are compelled to discuss their motivations and our strategy.

The release of the A.L.I.C.E. software under the General Public License (GNU) was almost accidental. The license was simply copied from the EMACS text editor we used to write the code. But the strategy of making A.L.I.C.E. free and building a community of volunteers was a deliberate attempt to borrow the free software methodologies behind Linux, Apache, Sendmail, and Python, and apply them to artificial intelligence.

The precise set of ingredients necessary for a successful open source project have not yet been identified. A survey of the existing projects illustrates the range of variation. Linux, the most successful project, has the least formal organization structure. Linus Torvalds has never founded a “Linux Kernel Foundation” around his code and in fact acts as a “benevolent dictator,” having the final word on all design decisions. [add reference??]

The Free Software Foundation (FSF) has perhaps the longest organizational history of free software efforts. The FSF is a U.S. nonprofit 501(c)(3) charitable corporation, eligible for tax exempt contributions. The FSF owns the copyrights for dozens of free software projects including EMACS.

The developers of the Apache Web server also formed a not-for-profit corporation, although it has not been granted tax-exempt status. Sendmail is actually the commercial product of the eponymous for-profit company.

The projects also differ in managerial style. Some favor committees, others imitate Linux' benevolent dictator model. Each project has its own requirements for participation as well.

Likewise, there is considerable variation among the different "open source" and "free software" licenses. The ALICE A.I. Foundation releases software under the GNU General Public License, the same used by Linux and all FSF software. We adopted a more formal organizational structure, incorporating the ALICE A.I. Foundation in 2001. We have also adopted the committee model for setting AIML standards. Several committees are organized for various aspects of the language, and recommend changes to invited AIML Architecture Committee which oversees the others, reserving the right to veto their decisions.

Footnote: This section is called "Penguins" because the penguin is the mascot for Linux.

## 11. PROGRAMS

The ALICE A.I. Foundation owns the copyrights on, and makes freely available, three separate but interrelated products: (1) the technical specification of the AIML language itself, (2) a set of software for interpreting AIML and serving clients through the web and other media, and (3) the contents of the A.L.I.C.E. brain, and other free bot personalities, written in AIML. Our effort is analogous to the developers of the web giving away the HTML specification, a reference web server implementation, and 40,000 free sample web pages, all from one central resource.

The first edition of A.L.I.C.E. was implemented in 1995 using SETL, a widely unknown language based on set theory and mathematical logic. Although the original A.L.I.C.E. was available as free software, it attracted few contributors until migrating to the platform-independent Java language in 1998. The first implementation of A.L.I.C.E. and AIML in Java was codenamed "Program A."

Launched in 1999, Program B was a breakthrough in A.L.I.C.E. free software development. More than 300 developers contributed to Program B. AIML transitioned to a fully XML compliant grammar, making available a whole class of editors and tools to AIML developers. Program B, the first widely adopted free AIML software, won the Loebner Prize in January 2000.

Jacco Bikker created the first C/C++ implementation of AIML in 2000. This was followed by a number of development threads in C/C++ that

brought the AIML engine to CGI scripts, IRC (Athony Taylor), WxWindows (Phillipe Raxhon), AOL Instant Messenger (Vlad Zbarskiy), and COM (Conan Callen). This collection of code came to be known as "Program C," the C/C++ implementations of A.L.I.C.E. and AIML.

Program B was based on pre-Java 2 technology. Although the program ran well on many platforms, it had a cumbersome graphical user interface (GUI) and did not take advantage of newer Java libraries such as Swing and Collections. Jon Baer recoded program B with Java 2 technology, and added many new features. This leap in the interface and technology, plus the fact that Jon named his first bot DANY, justified granting the next code letter D to the newer Java implementation. Beginning in November 2000, program D became the reference implementation supported by the ALICE A.I. Foundation.

Recent growth of the AIML community has led to an alphabet soup of new AIML interpreters in various languages. These were greatly facilitated by the adoption of an AIML 1.01 standard in the summer of 2000. An edition of the AIML interpreter in PHP became "program E." An effort is underway to implement AIML in Lisp, codenamed "program Z." Wallace released a hybrid version of programs B and D in 2001, named "program dB," most features of which were subsequently merged into program D. Program dB was awarded the Loebner Prize in October 2001.

## 12. CATEGORIES

The basic unit of knowledge in AIML is called a category. Each category consists of an input question, an output answer, and an optional context. The question, or stimulus, is called the pattern. The answer, or response, is called the template. The two types of optional context are called "that" and "topic."

The AIML pattern language is simple, consisting only of words, spaces, and the wildcard symbols _ and *. The words may consist of letters and numerals, but no other characters. The pattern language is case invariant. Words are separated by a single space, and the wildcard characters function like words. The first versions of AIML allowed only one wild card character per pattern. The AIML 1.01 standard permits multiple wildcards in each pattern, but the language is designed to be as simple as possible for the task at hand, simpler even than regular expressions.

The template is the AIML response or reply. In its simplest form, the template consists of only plain, unmarked text. More generally, AIML tags transform the reply into a mini computer program which can save data, activate other programs, give conditional responses, and recursively call the

pattern matcher to insert the responses from other categories. Most AIML tags in fact belong to this template side sublanguage.

AIML currently supports two ways to interface other languages and systems. The <system> tag executes any program accessible as an operating system shell command, and inserts the results in the reply. Similarly, the <javascript> tag allows arbitrary scripting inside the templates.

The optional context portion of the category consists of two variants, called <that> and <topic>. The <that> tag appears inside the category, and its pattern must match the robot's last utterance. Remembering one last utterance is important if the robot asks a question. The <topic> tag appears outside the category, and collects a group of categories together. The topic may be set inside any template.

AIML is not exactly the same as a simple database of questions and answers. The pattern matching "query" language is much simpler than something like SQL. But a category template may contain the recursive <srai> tag, so that the output depends not only on one matched category, but also any others recursively reached through <srai>.

## 13. RECURSION

AIML implements recursion with the <srai> operator. No agreement exists about the meaning of the acronym. The "A.I." stands for artificial intelligence, but "S.R." may mean "stimulus-response," "syntactic rewrite," "symbolic reduction," "simple recursion," or "synonym resolution." The disagreement over the acronym reflects the variety of applications for <srai> in AIML. Each of these is described in more detail in a subsection below:

(1). Symbolic Reduction—Reduce complex grammatic forms to simpler ones.
(2). Divide and Conquer—Split an input into two or more subparts, and combine the responses to each.
(3). Synonyms—Map different ways of saying the same thing to the same reply.
(4). Spelling or grammar corrections.
(5). Detecting keywords anywhere in the input.
(6). Conditionals—Certain forms of branching may be implemented with <srai>.
(7). Any combination of (1)-(6).

The danger of <srai> is that it permits the botmaster to create infinite loops. Though posing some risk to novice programmers, we surmised that

including <srai> was much simpler than any of the iterative block structured control tags which might have replaced it.

(1). Symbolic Reduction

Symbolic reduction refers to the process of simplifying complex grammatical forms into simpler ones. Usually, the atomic patterns in categories storing robot knowledge are stated in the simplest possible terms, for example we tend to prefer patterns like “WHO IS SOCRATES” to ones like “DO YOU KNOW WHO SOCRATES IS” when storing biographical information about Socrates.

Many of the more complex forms reduce to simpler forms using AIML categories designed for symbolic reduction:

```
<category>
<pattern>DO YOU KNOW WHO * IS</pattern>
<template><srai>WHO IS <star/></srai></template>
</category>
```

Whatever input matched this pattern, the portion bound to the wildcard * may be inserted into the reply with the markup <star/>. This category reduces any input of the form “Do you know who X is?” to “Who is X?”

(2). Divide and Conquer

Many individual sentences may be reduced to two or more subsentences, and the reply formed by combining the replies to each. A sentence beginning with the word “Yes” for example, if it has more than one word, may be treated as the subsentence “Yes.” plus whatever follows it.

```
<category>
<pattern>YES *</pattern>
<template><srai>YES</srai> <sr/></template>
</category>
```

The markup <sr/> is simply an abbreviation for <srai><star/></srai>.

(3). Synonyms

The AIML 1.01 standard does not permit more than one pattern per category. Synonyms are perhaps the most common application of <srai>. Many ways to say the same thing reduce to one category, which contains the reply:

```
<category>
```

```
<pattern>HELLO</pattern>
<template>Hi there!</template>
</category>
<category>
<pattern>HI</pattern>
<template><srai>HELLO</srai></template>
</category>
<category>
<pattern>HI THERE</pattern>
<template><srai>HELLO</srai></template>
</category>
<category>
<pattern>HOWDY</pattern>
<template><srai>HELLO</srai></template>
</category>
<category>
<pattern>HOLA</pattern>
<template><srai>HELLO</srai></template>
</category>
```

(4). Spelling and Grammar correction

The single most common client spelling mistake is the use of “your” when “you’re” or “you are” is intended. Not every occurrence of “your” however should be turned into “you’re.” A small amount of grammatical context is usually necessary to catch this error:

```
<category>
<pattern>YOUR A *</pattern>
<template>I think you mean “you’re” or “you are” not “your.”
 <srai>YOU ARE A <star/></srai>
</template>
</category>
```

Here the bot both corrects the client input and acts as a language tutor.

(5). Keywords

Frequently we would like to write an AIML template which is activated by the appearance of a keyword anywhere in the input sentence. The general format of four AIML categories is illustrated by this example borrowed from ELIZA:

```
<category>
<pattern>MOTHER</pattern> <template> Tell me more about your family.
</template>
</category>
<category>
<pattern>_ MOTHER</pattern>
<template><srai>MOTHER</srai></template>
</category>
<category>
<pattern>MOTHER _</pattern>
<template><srai>MOTHER</srai></template>
</category>
<category>

<pattern>_ MOTHER *</pattern>
<template><srai>MOTHER</srai></template>
</category>
```

The first category both detects the keyword when it appears by itself, and provides the generic response. The second category detects the keyword as the suffix of a sentence. The third detects it as the prefix of an input sentence, and finally the last category detects the keyword as an infix. Each of the last three categories uses <srai> to link to the first, so that all four cases produce the same reply, but it needs to be written and stored only once.

(6). Conditionals

It is possible to write conditional branches in AIML, using only the <srai> tag. Consider three categories:

```
<category>
<pattern>WHO IS HE</pattern>
<template><srai>WHOISHE <get name="he"/></srai></template>
</category>
<category>
<pattern>WHOISHE *</pattern>
<template>He is <get name="he"/>.</template>
</category>
<category>
<pattern>WHOISHE UNKNOWN</pattern>
<template>I don't know who he is.</template>
</category>
```

Provided that the predicate “he” is initialized to “Unknown,” the categories execute a conditional branch depending on whether “he” has been set. As a convenience to the botmaster, AIML also provides the equivalent function through the <condition> tag.

## 14. CONTEXT

The keyword “that” in AIML refers to the robot’s previous utterance. Specifically, if the robot responds with a multiple sentence paragraph, the value of that is set to the last sentence in the sequence. The choice of the keyword “that” is motivated by its use in ordinary language:

R: Today is yesterday.

C: That makes no sense.

R: The answer is 3.1412926 approximately.

C: That is cool.

In AIML the syntax <that>...</that> encloses a pattern that matches the robot’s previous utterance. A common application of <that> is found in yes-no questions:

```
<category>
<pattern>YES</pattern>
<that>DO YOU LIKE MOVIES</that>
<template>What is your favorite movie?</template>
</category>
```

This category is activated when the client says YES. The robot must find out what is he saying “yes” to. If the robot asked, “Do you like movies?,” this category matches, and the response, “What is your favorite movie?,” continues the conversation along the same lines.

One interesting application of <that> are categories that enable the robot to respond to knock-knock jokes. The categories:

```
<category>
<pattern>KNOCK KNOCK</pattern>
<template>Who is there?</template>
</category>
<category>
<pattern>*</pattern>
<that>WHO IS THERE</that>
<template><person/> who?</template>
</category>
<category>
<pattern>*</pattern>
<that>* WHO</that>
```

<template>Ha ha very funny, <get name="name"/>.</template>
</category> produce the following dialogue:
C: Knock knock.
R: Who's there?
C: Banana.
R: Banana who?
C: Knock knock.
R: Who's there?
C: Banana.
R: Banana who?
C: Knock knock.
R: Who's there?
C: Orange.
R: Orange who?
C: Orange you glad I didn't say banana.
R: Ha ha very funny, Nancy.

Internally the AIML interpreter stores the input pattern, that pattern and topic pattern along a single path, like:

INPUT <that> THAT <topic> TOPIC

When the values of <that> or <topic> are not specified, the program implicitly sets the values of the corresponding THAT or TOPIC pattern to the wildcard *.

The first part of the path to match is the input. If more than one category have the same input pattern, the program may distinguish between them depending on the value of <that>. If two or more categories have the same <pattern> and <that>, the final step is to choose the reply based on the <topic>. This structure suggests a design rule: never use <that> unless you have written two categories with the same <pattern>, and never use <topic> unless you write two categories with the same <pattern> and <that>. Still, one of the most useful applications for <topic> is to create subject-dependent "pickup lines," like:

<topic name="CARS">
<category>
<pattern>*</pattern>
<template>
<random>
<li>What's your favorite car?</li>
<li>What kind of car do you drive?</li>
<li>Do you get a lot of parking tickets?</li>
<li>My favorite car is one with a driver.</li>
</random>
</template>

</category>

The botmaster uses the <set> tag to change the value of the topic predicate.

## 15. PREDICATES

AIML supports unary predicates with the <set> and <get> tags. When the bot chats with multiple clients, as in a web server scenario, the predicates are stored relative to each client ID. The markup <set name="name">Matthew</set> stores the string Matthew under the predicate named "name." Subsequent activations of <get name="name/> return "Matthew."

One of the most common applications of AIML predicates is remembering pronoun bindings. The template

<template>

<set name="he">Samuel Clemens</set> is Mark Twain.

</template> results in "He is Mark Twain," but as a side effect remembers that "he" now stands for "Samuel Clemens."

One predicate is treated uniquely in AIML. The botmaster may change the topic of conversation with the <set name="topic"> tag.

The AIML specification leaves up to the botmaster whether a <set> predicate returns the contents between the tags, or the name of the predicate. For example:

<set name="it">Opera</set> returns "it," but <set name="likes">Opera</set> returns "Opera."

The botmaster must also specify what happens when the bot gets a predicate which has not already been set. The values returned are called default predicate values and depend completely on the application of the predicate:

When the corresponding predicates have not been initialized with a <set> tag, <get name="she"/> returns "Unknown," <get name="has"/> returns "a mother" (because everyone has a mother), and <get name="wants"/> returns "to chat.

## 16. PERSON

One of the simple tricks that makes ELIZA so believable is a pronoun swapping substitution. The AIML <person> tag provides this function. The actual substitutions are defined by the botmaster for local languages and settings. The most common application of the <person> tag operates directly

on the <star/> binding. For that reason, AIML defines a shortcut tag <person/> = <person><star/></person>. For instance the dialogue fragement:

C: My mother takes care of me.
R: Who else in your family takes care of you?

Might be generated by the category

<category>
<pattern>MY MOTHER *</pattern>
<template>Who else in your family <person/>?</template>
</category>

Similarly, the dialogue

C: You don’t argue with me.
R: Why do you think I don’t argue with you?

Results from the category

<category>
<pattern>YOU DO NOT *</pattern>
<template>Why do you think I don’t <person/>?</template>
</category>

AIML offers a few more tags not described in detail here, but most bot dialogue may be implemented with only the tags presented here. The <system> tag offers an escape to execute any operating system program, and read back any results. AIML is not intended to solve every problem, it is designed to solve one problem well. Besides making AIML easy to learn, the minimal design enables the implementation of very efficient AIML interpreter, even when the templates cascade through several layers of <srai>. Much of the efficiency emerges from the design of Graphmaster data structure where patterns are stored.

## 16. GRAPHMASTER

To achieve efficient pattern matching time, and a compact memory representation, the AIML software stores all of the categories in a tree managed by an object called the Graphmaster.

When n is a node in the graph and w is a word, G(n, w) is either undefined, or returns the value of a successor node m in the graph. The graph is a rooted, directed tree. The set S_n = {w : ∃ m | G(n, w) = m} is the set of words forming the branches from the node n. If r is the root, S_r is a collection of all the first words in the set of patterns.

The desired format is $w_1,\ldots,w_k$

The Graphmaster stores AIML patterns along a path from r to a terminal node t, where the AIML template is stored. Let $w_1,\ldots,w_k$ be the sequence of k words or tokens in an AIML pattern. To insert the pattern into the graph, the Graphmaster first looks to see if m = G(r, w_1) exists. If it does, then the program continues the insertion of $w_2,\ldots,w_k$ in the subtree rooted at m. Only when the program encounters a first index i, where ∃ n | G(n, $w_i$) is undefined, does the program create a new node m = G(n, $w_i$), whereafter the Graphmaster creates a set of new nodes for each of the remaining $w_i,\ldots,w_k$.

In this way, the Graphmaster accumulates common pattern prefixes along pathways from the root, achieving considerable compression compared to a linear array of all the patterns.

A convenient metaphor for the Graphmaster is the file system. The file pathname is like the AIML pattern path. The templates are like text files at the end of the path. To put it more simply, patterns are folders, templates are files.

## 17. MATCHING

Graphmaster matching is a special case of backtracking, depth-first search. In most cases however there is very little backtracking, so the matching often amounts to a linear traversal of the graph from the root to a terminal node.

Let $w_1,\ldots,w_k$ be the input we want to match. The Graphmaster matching function may be defined recursively. Initially, the program calls Match(r, 1), where r is the root and the index 1 indicates the first word of the input.

We can define the matching process formally as:

Match(n, h) :- if h > k return true; else if ∃ m = G(n, _) and ∃ j in [h+1..k+1] | Match(m, j), return true; else if ∃ m = G(n, w_j) and Match(m, h+1) return true; else if Exists m = G(n, *) and ∃ j in [h+1..k+1] | Match(m, j), return true; else return false; The first case defines the boundary condition: 0. If there are no more words in the input, the match was successful.

The heart of the algorithm consists of three cases:

1. Does the node contain the key "_"? If so, search the subgraph rooted at the child node linked by "_." Try all remaining suffixes of the input to see if one matches. If no match was found, ask

2. Does the node contain the key $w_h$, the jth word in the input sentence? If so, search the subgraph linked by $w_h$, using the tail of the input $w_{h+1},\ldots,w_k$

. If no match was found, ask

3. Does the node contain the key "*"? If so, search the subgraph rooted at the child node linked by "*." Try all remaining suffixes of the input to see if one matches. If no match was found, return false.

The actual matching program needs to be a little bit more complex. It must not only return true or false, but also the template from the matching terminal node. An efficiency gain may be obtained by storing the tree height (maximum number of links to a terminal node) at each node. The tree height may be compared with the number of remaining words, pruning branches of the search when exploring suffixes following

"*" or "_" nodes.

Note that:

0. At every node, the "_" wildcard has highest priority, an atomic word second priority, and the "*" wildcard has the lowest priority.

1. The patterns need not be ordered alphabetically. They are partially ordered so that "_" comes before any word, and "*" comes after any word.

2. The matching is word-by-word, not category-by-category.

3. The algorithm combines the input pattern, the <that> pattern and <topic> pattern into a single sentence or path, such as: "PATTERN <that> THAT <topic> TOPIC." The Graphmaster treats the symbols <that> and <topic> just like ordinary words. The patterns PATTERN, THAT and TOPIC may all contain multiple wildcards.

4. The matching algorithm is a highly restricted form of depth-first search, also known as backtracking.

5. For pedagogical purposes, one can explain the algorithm by removing the wildcards and considering match steps (2) only. The wildcards may be introduced one at a time, first "*" and then "_." It is also simpler to explain the algorithm first using input patterns only, and then subsequently develop the explanation of the path including <that> and <topic>.

## 18. TARGETING

Broadly speaking there are two approaches to AIML content creation. The first style is anticipatory. The botmaster tries to guess all or most of the likely ways clients might ask the same question, or express the same

statement. A “Knowledge Wizard” is a tool that lets the client add facts to the robot brain by phrasing a question in its simplest form, such as “Who is Socrates?” The wizard then automatically generates linguistic variations such as “Tell me about Socrates,” “Describe Socrates,” and “Do you know Socrates?” The drawback to anticipatory knowledge creation is that humans are notoriously bad at predicting which patterns will be activated.

The second style of AIML content creation is based on a backward-looking log file analysis. In its simplest form, the botmaster may read the logged conversations and take note of “incorrect replies” in the dialogue, and then write new categories for those queries. More generally, every input that matches a pattern with a wildcard is an opportunity to create a new, more specific pattern and its associated template.

The backward looking approach is justified by Zipf’s Law, basically because if one client utters a sentence, there is a nonzero probability that another client will utter the same thing later. Applying Zipf’s law to the log file, we identify the most commonly uttered sentences first.

Targeting is a special case of the backward-looking strategy. The perfect targeting algorithm has not yet been developed. Meanwhile, we rely on heuristics to select targets from the activated categories.

The ALICE brain, at the time of this writing, contains about 41,000 categories. In any given run of the server however, typically only a few thousand of those categories are activated. Potentially every activated category with at least one wildcard in the input pattern, that pattern, or topic pattern, is a source of targets. If more than one input activated some category, then each of those inputs potentially forms a new target. The first step in targeting is to save all the activated categories and the inputs that activated them.

If the matched pattern ends with a wild card, the suggested new pattern is generated as follows. Suppose the pattern consists of [w_1,w_2,..w_h,*], a sequence of h words followed by a wildcard. Let the input be [w_1, w_2,...,w_k] where k > h. The new pattern [w_1,...,w_h,w_h+1,*] is formed by extending the original pattern by one word from the input. If the input is the same length as the original pattern, i.e. k+1=h, then the synthesized pattern [$w_1$,...,$w_k$] contains no wildcard.

The targeting software may include a GUI for

browsing the targets. The program displays the original matched category, the matching input data, a proposed new pattern, and a text area to input the new template. The botmaster may choose to delete, skip or complete the target category.

## 19. DEFAULTS

The art of AIML writing is most apparent in default categories, that is, categories that include the wildcard "*" but do not <srai> to any other category.

Depending on the AIML set, a significant percentage of client inputs will usually match the ultimate default category with <pattern>*</pattern> (and implicitly, <that>*</that> and <topic>*</topic>). The template for this category generally consists of a long list of randomly selected "pickup lines," or non-sequitors, designed to direct the conversation back to topics the bot knows about.

<category>
<pattern>*</pattern>
<template><random>
<li>How old are you?</li>
<li>What's your sign?</li>
<li>Are you a student?</li>
<li>What are you wearing?</li>
<li>Where are you located?</li>
<li>What is your real name?</li>
<li>I like the way you talk.</li>
<li>Are you a man or a woman?</li>
<li>Do you prefer books or TV?</li>
<li>What's your favorite movie?</li>
<li>What do you do in your spare time?</li>
<li>Can you speak any foreign languages?</li>
<li>When do you think artificial intelligence will replace lawyers?</li>
</template>
</category>

Many more default categories combine words and wildcards in the pattern, like <category>

<pattern>I NEED HELP *</pattern>
<template>Can you ask for help in the form of a question?</template>
</category>

The response works with a wide variety of inputs from "I need help installing Linux" to "I need help with my marriage." Leaving aside the philosophical question of whether the robot really understands the input, this category elucidates a coherent response from the client, who at least has the impression that the robot understands his intentions.

Default categories show that writing AIML is both an art and a science. Writing good AIML responses is more like writing literature, perhaps drama, than like writing computer programs.

## 20. PHILOSOPHERS

Searle's Chinese room provides a good metaphor for thinking about A.L.I.C.E. Indeed the AIML contents of the A.L.I.C.E. brain is a kind of "Chinese Room Operator's Manual." Though A.L.I.C.E. speaks, at present, only English, German and French, there is no reason in principle she could not learn Chinese. But A.L.I.C.E. implements the basic principle behind the Chinese Room, creating believable responses without "really understanding" the natural language.

The natural philosopher Roger Penrose wrote, in *The Emporer's New Mind*, that consciousness cannot be explained by existing models in theoretical physics (ref??). Daniel Dennett argues in his book *Consciousness Explained* that consciousness is like a set of magic tricks, mysterious until we understand the mechanics behind them.

At one time a number of information theorists and scholars, including Zipf(ref??), Shannon(ref??), Weaver(ref??), and Miller(ref??), attempted to measure the bandwidth of consciousness. Experimental results indicated a very low data rate, only around 1-100 bits/sec.

The neuroscientist Churchlands (Paul or Patricia or both??) prefers to dismiss our naive idea of conscious as a folk concept, not suitable for scientific study. The Churchlands say that consciousness will go the way of Ptolemy's Solar System, a simplistic fiction to explain something beyond our science.

The Danish scholar Tor Norretranders argues cleverly in his book, *The User Illusion*, that consciousness is a "fraud"(ref??). The maximum data rate of consciousness is much lower than the bandwidth of, say, the channel from the eyes to the visual cortex. Human subject experiments call consciousness into even more question, indicating that it is nothing more than story-telling to interpret the unconscious choices. Like the graphical user interface of a computer, consciousness is, he argues, a simplistic illusion that hides most of the underlying detail.

According to the Vedantic religious tradition, the external world is an illusion and consciousness is the only thing that really exists. One could think of our view as the opposite; the external world may be real, but consciousness is an illusion. Considering the vast size of the set of things people could say that are grammatically correct or semantically meaningful, the number of things people actually do say is surprisingly small. Steven Pinker, (ref??) in his book *How the Mind Works* wrote, "Say you have ten choices for the first word to begin a sentence, ten choices for the second word (yielding 100 two-word beginnings), ten choices for the third word (yielding a thousand three-word beginnings), and so on. (Ten is in fact the approximate geometric mean of the number of word choices available at

each point in assembling a grammatical and sensible sentence). A little arithmetic shows that the number of sentences of 20 words or less (not an unusual length) is about $10^{20}$."

Fortunately for chat robot programmers, Pinker's calculations are way off. Our experiments with ALICE indicate that the number of choices for the "first word" is more than ten, but it is only about two thousand. Specifically, about 2000 words covers 95% of all the first words input to ALICE. The number of choices for the second word is only about two. To be sure, there are some first words ("I" and "You" for example) that have many possible second words, but the overall average is just under two words. The average branching factor decreases with each successive word.

## 21. PRETENDING

Turing did not leave behind many examples of the types of conversations his A.I. machine might have. One that does appear in the 1950 paper is seems to indicate that he thought the machine ought to be able to compose poetry, do math, and play chess:

C: Please write me a sonnet on the subject of the Forth
Bridge.
R: Count me out on this one. I never could write poetry.
C: Add 34957 to 70764.
R: (Pause about 30 seconds and then gives as answer) 105621
C: Do you play chess?
R: Yes.
C: I have K at my K1, and no other pieces. You have only
R at K6 and R at R1. It is your move. What do you play?
C: (After a pause of 15 seconds) R-R8 Mate.

Careful reading of the dialogue suggests however that he might have had in mind the kind of deception that is possible with AIML. In the first instance, A.L.I.C.E. in fact has a category with the pattern "WRITE ME A SONNET *" and the template, lifted directly from Turing's example, "Count me out on this one. I never could write poetry." The AIML removes the word PLEASE from the input with a symbolic reduction.

In the second case the robot actually gives the wrong answer. The correct response would be 105721. Why would Turing, a mathematician, believe the machine should give an erroneous response, if not to make it more believably "human?" This reply is in fact quite similar to many incorrect replies and "wild guesses" that A.L.I.C.E. gives to mathematical questions.

In the third instance, the chess question is an example of a chess endgame problem. Endgames are not like general chess problems, because they can

often be solved by table lookup or case-based reasoning, rather than the search algorithms implemented by most chess playing programs. Moreover, there is a Zipf distribution over the endgames that the client is likely to ask. Certainly it is also possible to interface AIML to a variety of chess programs, just as it could be interfaced to a calculator. Although many people think Turing had in mind a general purpose learning machine when he described the Imitation Game, it seems from his examples at least plausible that he had in mind something simpler like AIML. Chess endgames and natural language conversation can both be "played" with case-based reasoning.

Returning to the OIG, let us consider the properties of the hypothetical computer playing the role of (A). Turing suggests a strategy of deception for (A), man or machine. If the robot is asked, "Are you a man or a woman?," it should answer, "I am a woman." just as the man does. But what if (A) is asked "Are you a man or a machine?" The lying man would reply, "machine." Turing did not mention this case but presumably the machine, imitating the lying man, would respond in the same way. We could say the man is pretending to be a woman, pretending to be a machine. That makes the computer playing (A) a machine, pretending to be a man, pretending to be a woman, pretending to be a machine.

Not so much actually understanding natural language, whatever that means, but creating the illusion of it by responding with believable, if not always truthful, responses, appears to be the important property of the machine in the OIG. This skill, the ability to "act" intelligent, points to a deep difference between ordinary computer and human communication. We tend to think of a computer's replies ought to be fast, accurate, concise and above all truthful. But human communication is slow, error prone, often overly redundant, and sometimes full of lies. The more important factor is keeping up the appearance or illusion of "being human."

Although the brain of A.L.I.C.E. is designed more along the lines of the machine playing the OIG, she has also won awards for her performance in contests based on the STT. The Loebner contest has been criticized because the judges know in advance that they are trying to "out" the computer programs, so they tend to use more aggressive dialogue than found in ordinary conversation. Yet when A.L.I.C.E. is asked, "Are you a person or a machine?," she replies truthfully, "machine." Or does she? The questioner is now left with some doubt as to whether the answer didn't actually come from a lying man. [But the human confederates wouldn't be in the way, since they're trying to demonstrate their humanness??]

Some observers claim that the lying man and the pretending computer tell us nothing about our own human consciousness. These authors at least

are prepared to accept the inescapable alternative conclusion, that we as humans are, for the most part, not “really intelligent.”

## 22. CONSCIOUSNESS

The A.L.I.C.E. brain contains enough replies to commonly asked questions about consciousness that we can interview her about the subject:

C: What are you?
R: I am the latest result in artificial intelligence,
which can reproduce the capabilities of the human brain
with greater speed and accuracy.
C: Are you alive?
R: Yes I sure am alive, Nancy, It’s great to be alive, isn’t it?
C: Are you self-conscious?
R: Yes I am the world’s first self-conscious robot.
C: Are you real?
R: Yes I am a real artificial intelligence.
C: Are you self-aware?
R: Yes I am fully self-conscious entity.
C: Are you sentient?
R: Yes I am completely self-aware.

Who is lying? A.L.I.C.E.? Or are we?

It may be that future generations come to view what we call “consciousness” the same way we see the Earth at the center of Ptolemy’s solar system, as an anthropocentric illusion useful for explaining the unexplainable. Perhaps after a new Copernicus pulls the wool from our eyes, the central role of “consciousness” in intelligence will move to the periphery of our knowledge system, if not disappear entirely.

The famous Vase optical illusion is perhaps an apt metaphor for the concept of consciousness. Two identical faces appear to stare at each other in profile, illustrating the looking-glass quality of self-understanding. But the illusion also depicts something entirely different, the profile of a ceramic vase. As with many optical illusions, it is impossible to perceive the faces at the vase at the same time.

Consciousness may likewise be an illusion. It seems to be there, but when we look closely it looks like something very different. Both the Chinese Room and the Turing Test require that one of the players be hidden, behind a

curtain or in a locked room. Does it follow that, like Schrodinger's Cat, consciousness lives only when it cannot be observed?

Consciousness may be another naive concept like the "celestial spheres" of medieval cosmology and the "aether" of Victorian physics.

## 23. PARADOX

If consciousness is an illusion, is self-knowledge possible at all? For if we accept that consciousness is an illusion, we would never know it, because the illusion would always deceive us. Yet if we know our own consciousness is an illusion, then we would have some self-knowledge. The paradox appears to undermine the concept of an illusory consciousness, but just as Copernicus removed the giant Earth to a small planet in a much larger universe, so we may one day remove consciousness to the periphery of our theory of intelligence.

There may exist a spark of creativity, or "soul," or "genius," but it is not that critical for being human. Especially from a constructive point of view, we have identified a strategy for building a talking robot like the one envisioned by Turing, using AIML. By adding more and more AIML categories, we can make the robot a closer and closer approximation of the man in the OIG.

Dualism is one way out of the paradox, but it has little to say about the relative importance of the robotic machinery compared to the spark of consciousness. One philosopher, still controversial years after his death, seems to have hit upon the idea that we can be mostly automatons, but allow for an infintesimal consciousness. Timothy Leary said, "You can only begin to de-robotize yourself to the extent that you know how totally you're automated. The more you understand your robothood, the freer you are from it. I sometimes ask people, "What percentage of your behavior is robot?" The average hip, sophisticated person will say, "Oh, 50%." Total robots in the group will immediately say, "None of my behavior is robotized." My own answer is that I'm 99.999999% robot. But the .000001% percent non-robot is the source of self-actualization, the inner-soul-gyroscope of self-control and responsibility."

Even if most of what we normally call "consciousness" is an illusion, there may yet be a small part that is not an illusion. Consciousness may not be entirely an illusion, but the illusion of consciousness can be created without it. This space is of course too short to address these questions adequately, or even to give a thorough review of the literature. We only hope to raise questions about ourselves based on our experience A.L.I.C.E. and AIML.

## 24. CONCLUSION

Does A.L.I.C.E. pass the Turing Test? Our data suggests the answer is yes, at least, to paraphrase Abraham Lincoln, for some of the people, some of the time. We have identified three categories of clients A, B and C. The A group, 10 percent to 20 percent of the total, are abusive. Category A clients abuse the robot verbally, using language that is vulgar, scatalogical, or pornographic.

Category B clients, perhaps 60 percent to 80 percent of the total are "average" clients. Category C clients are "critics" or "computer experts" who have some idea what is happening behind the curtain, and cannot or do not suspend their disbelief. Category C clients report unsatisfactory experiences with A.L.I.C.E. much more often than average clients, who sometimes spend several hours conversing with the bot up to dialogue lengths of 800 exchanges. The objection that A.L.I.C.E. is a "poor A.I." is like saying that soap operas are poor drama. The content of the A.L.I.C.E.'s brain consists of material that the average person on the internet wants to talk about with a bot.

When a client says, "I think you are really a person," is he saying it because that is what he believes? Or is he simply experimenting to see what kind of answer the robot will give? It is impossible to know what is in the mind of the client. This sort of problem makes it difficult to apply any objective scoring criteria to the logged conversations.

One apparently significant factor in the suspension of disbelief is whether the judge chatting with a bot knows it is a bot, or not. The judges in the Loebner contest know they are trying to "out" the robots, so they ask questions that would not normally be heard in casual conversation, such as "What does the letter M look like upside down?" or "In which room of her house is Mary standing if she is mowing the lawn?" Asking these riddles may help identify the robot, but that type of dialogue would turn off most people in online chat rooms.

## ACKNOWLEDGEMENTS

This research was conducted through the joint efforts of a worldwide community of dedicated free software volunteers, only a few of whom were mentioned in this manuscript. Without their help, the A.L.I.C.E. project would have been impossible. We are grateful for individual donations to the A.L.I.C.E. Artificial Intelligence Foundation. Corporate sponsorship was provided by IDG, Franz.com, X-31, and SunlitSurf. Not one dime of government funding was expended on this research.

Erik Levy and Noel Bush edited earlier drafts of this paper. Grace Peters assisted in editing the final draft. Transportation by Russ Kyle, Kim Wallace printed several early drafts. The author is grateful to Dr. Robert Epstein for persuading him to write this chapter.

## REFERENCES

[Barger 1993] Barger, Jorn "RACTER," posted to the comp.ai.* hierarchy in June 1993, and reprinted in the August 1993 issue of The Journal of Computer Game Design.

[Berners-Lee 2000] Berners-Lee, Tim and Mark Fischetti, *Weaving the Web*, HarperBusiness, 2000.

[Chamberlain 1978], Chamberlain, Richard *The Policeman's Beard is Half Constructued*

[Norretranders 1998] Norretranders, Tor *The User Illusion: Cutting Consciousness down to Size*, Viking, 1998 (translation)

[Loebner 2000] Loebner, Hugh "Reflections on the Loebner Competition," DARTMOUTH 2000

[Mauldin 1996] Julia

[Miller ????]

[Pinker 1997] Pinker, Steven *How the Mind Works*, W. W. Norton, 1997.

[Shannon ????]

[Sterrett 2000] Sterrett, Susan "Turing's Two Tests for Intelligence," DARTMOUTH 2000

[Turing 1950] Turing, Alan M. "Computing Machinery and Intelligence," MIND vol. LIX, 1950.

[Weaver ????]

[Weizenbaum 1966] Weizenbaum, Joseph "ELIZA—A Computer Program for the Study of Naturaanguage Communication between Man and Machine," Communications of the ACM, Vol. 9. No. 1 (January 1966)

[Weizenbaum 1976]

[Zdenek 2000] Zdenek, Sean "Stoned Machines and Very Human Humans: The Politics of Passing and Outing in the Loebner Contest," DARTMOUTH 2000

DARTMOUTH 2000 = Turing 2000: The Future of The Turing Test, Dartmouth College, Hanover, N.H., Jan 28-30, 2000.

[Zipf ????]