М.С. ВЕНСКОВИЧ, Л.Н. ШПУДЕЙКО, Н.В. КИСТЕНЬ

ARTIFICIAL INTELLIGENCE

СБОРНИК ТЕКСТОВ ПО ЧТЕНИЮ НА АНГЛИЙСКОМ ЯЗЫКЕ

Education

UNIVERSITY

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

КАФЕДРА ИНОСТРАННЫХ ЯЗЫКОВ ТЕХНИЧЕСКИХ СПЕЦИАЛЬНОСТЕЙ

ARTIFICIAL INTELLIGENCE

СБОРНИК ТЕКСТОВ ПО ЧТЕНИЮ НА АНГЛИЙСКОМ ЯЗЫКЕ

для студентов специальности
1- 40 03 01 Искусственный интеллект

Брест 2008

УДК 804.01-07

Сборник текстов по чтению на английском языке предназначается, прежде всего, для студентов специальности 1 – 40 03 01 Искусственный интеллект, но может использоваться и студентами, изучающими вычислительные машины, системы, комплексы и сети.

Цель пособия – совершенствование навыков чтения и понимания аутентичной научно–технической литературы по изучаемой специальности, развитие навыков перевода.

Составители: М.С. Венскович, доцент, к.ф.н.
Л.Н. Шпудейко, преподаватель
Н.В. Кистень, преподаватель

Рецензент: Я.В. Подрядчикова, кандидат филологических наук, доцент

CONTENTS

ПРЕДИСЛОВИЕ

Сборник текстов по чтению на английском языке предназначается прежде всего для студентов специальности 1 – 40 03 01 Искусственный интеллект, но может использоваться и студентами, изучающими вычислительные машины, системы, комплексы и сети.

Цель пособия – совершенствование навыков чтения и понимания аутентичной научно – технической литературы по изучаемой специальности; развитие навыков перевода.

Пособие состоит из трех основных разделов, включающих 36 текстов; 13 текстов для дополнительного чтения и словаря.

Тексты, представленные в первых трех разделах, тематически связаны между собой и касаются, главным образом, истории, развития и применения искусственного интеллекта и родственных с ним технологий, разработки новых усовершенствованных технологий компьютеризации.

Во всех текстах отдельные в смысловом отношении части несут порядковую нумерацию для упрощения обращения к тексту при выполнении языковых и/ или речевых учебных действий на его основе.

UNIT I. ARTIFICIAL INTELLIGENCE

What Exactly Is Artificial Intelligence?

Although most attempts to define complex and widely used terms end in futility, it is useful to draw at least an approximate boundary around the concept to provide a perspective on the course. To do this we take the by no means universally accepted *Definition:* Artificial Intelligence (AI): is the study of how to make computers do things which, at the moment, people do better. This definition is somewhat vague due to its reference to the current state of computer science. The ability to solve problems of one sort or another is widely used as a measure of intelligence in many different contexts. It is rather obvious that intelligent machines are unlikely to serve any practical purpose unless they are capable of coping with some of the myriad of simple (or not-so-simple) problems which people overcome as a matter of routine. Problems come in a bewildering variety of shapes and sizes. There are problems which can be solved with patience and perseverance, and others which require flair and intuition. There are formal, abstract problems, like those involved in game playing, the solution of which may be of little more than academic interest. There are many problems that are practical and urgent, matters of life and death even. Some problems yield to elementary common sense; others can only be solved with the help of obscure knowledge. There are several reasons one might want to model human performance at these sorts of tasks:

- To test psychological theories of human performance.
- To enable computers to understand human reasoning.
- To enable people to understand computer reasoning.
- To exploit what knowledge we can glean from people.

Intelligence

It is difficult to discuss the workings of even relatively simple machines, such as washing machines or sewing machines, unless we understand the functions they were designed to perform. Since artificial intelligence is concerned with perhaps the most complex kind of machines we can imagine, "intelligent machines", perhaps we should try to define what we expect such machines to do. Obviously, we expect them to be intelligent.

What do we mean by *intelligence?* The dictionary gives the following definition *Intelligence:* The faculty of understanding.

Understanding: To comprehend something, or to recognize its significance. This is a concept that seems clear enough. When we apply it subjectively, it seems to correspond reasonable accurately to our own individual experience of what it is like to be intelligent or to use our intelligence. Unfortunately, it begins to fall apart when we try to apply it objectively, to consider intelligence as a faculty which might be shared by other entities, whether living or mechanical.

The main problem is that we know what it feels like to understand something, and are generally willing to credit other people / things with sensations similar to our own. Take a simple example, a familiar piece of machinery, the thermostat in a central heating system. It does not just recognize when the temperature falls below or rises above a certain level; it responds by taking the appropriate action. In a single very limited respect it seems to possess understanding and to demonstrate this in the clearest possible fashion by behaving intelligently. If the thermostat is intelligent, we devalue the word to the point where it becomes meaningless.

General intelligence has turned out to be a concept of dubious value when applied in practice, and the whole question of using IQ tests to measure people's worth or suitability for a job has become extremely controversial. So should we break intelligence down into separate faculties such as perception, reason and creativity? If so, what is the difference between intelligence and knowledge?

Knowledge

One of the few hard and fast results to come out of the first three decades of AI research is that intelligence requires knowledge. To compensate for its one overpowering asset, indispensability, knowledge possesses some less desirable properties, including:

- It is volumous.
- It is hard to characterize accurately.
- It is constantly changing.
- It differs from data by being organized.

There are many different ways of categorizing knowledge types; one of the main distinctions is the difference between induced knowledge and deduced knowledge. This is best explained by means of the following example.

Consider a commonplace skill which most children master between the ages of five and ten - catching a ball. At the age of five a child may have difficulty in catching a beach ball gently tossed at a few yards, yet a few years later he/she will probably be able to catch a tennis ball lobbed high in the air from twenty yards away. Human beings are not capable of mastering the technique for calculating ballistic trajectories at such an impressively early age. The child's understanding has been gained by induction. It is as a result of watching the trajectories of many balls and trying to catch them that the child has been capable of predicting the trajectory of the next ball he/she wants to catch.

A computer system, on the other hand, would rely on information on the projectile velocity and trajectory to calculate the future location of a projectile using Newton's laws. This would be dependant on a rigorous and mathematically explicit set formula programmed into the computer. The program enables the computer to deduce the flight path of a projectile by reference to the set of formal mathematical rules.

Few people would dispute the proposition that calculating a ballistic trajectory mathematically requires more intelligence than being able to catch a ball. So there is an important distinction to be made between knowledge and intelligence. It should also be clear that a machine may store knowledge without necessarily possessing intelligence; an intelligent machine which had no knowledge is impossibility.

The question of how, to what extent, and in what sense can a machine be imbued with knowledge is thus fundamental to all aspects of artificial intelligence.

History

In the early 1900's, Torres y Quevedo, a Spanish inventor, built a machine that could checkmate its opponent with a rook and a king against a king. Systematic work began only after the invention of the digital computer. The 1st scientific article on AI was published by Alan Turing in 1950, and the 1st full time research group was started in 1954 at Carnegie Mellon University by Allen Newell and Herbert Simon. But the field of AI as a coherent area of research is roughly about 40 years old. It all started in the 1956 Dartmouth conference where ten young researchers had the same dream of using a computer to model the ways humans think. Their hypothesis was that mechanisms of human thought could be precisely modeled and simulated on a digital computer. This is what the whole foundation of AI is based on.

Within a few years AI seemed to take off. Checkers, translations from sentences to code into human understandable words, and identification of patterns were created. On the downside arguments arose. People were sure that technology would fail, and that getting computers to "think" was impossible. They confused the early difficulties and stumbles with the fundamental limits of technology. In the words of Scotty, "I can't do it, Captain. I just don't have the power." It was impossible for artificial intelligence to be created with the limited technology of the early 60's. Past failures and new technology led to many advances in science's history. Each failure added more information to build on.

What Are the Branches of Artificial Intelligence?

Here's a list, but some of the branches are surely missing, because no one has identified them yet. Some of these may be regarded as concepts or topics rather than full branches.

- *Logical AI*

What a program knows about the world in general the facts of the specific situation in which it must act, and its goals are all represented by sentences of some mathematical logical language. The program decides what to do by inferring that certain actions are appropriate for achieving its goals.

- *Search*

AI programs often examine large numbers of possibilities, e.g. moves in a chess game or inferences by a theorem proving program. Discoveries are continually made about how to do this more efficiently in various domains.

- *Pattern recognition*

When a program makes observations of some kind, it is often programmed to compare what it sees with a pattern. For example, a vision program may try to match a pattern of eyes and a nose in a scene in order to find a face. More complex patterns, e.g. in a natural language text, in a chess position, or in the history of some event are also studied. These more complex patterns require quite different methods than do the simple patterns that have been studied the most.

- *Representation*

Facts about the world have to be represented in some way. Usually languages of mathematical logic are used.

- *Inference*

Some facts can be inferred from others. Mathematical logical deduction is adequate for some purposes, but new methods of non-monotonic inference have been developed since the 1970s.

- *Common sense knowledge and reasoning*

This is the area in which AI is farthest from human-level, in spite of the fact that it has been an active research area since the 1950s. While there has been considerable progress, e.g. in developing systems of non-monotonic reasoning and theories of action, yet more new ideas are needed.

- *Learning from experience*

Programs do that. The approaches to AI based on connectionism and neural nets specialize in that. There is also learning of laws expressed in logic. Programs can only learn what facts or behaviour their formalisms can represent, and, unfortunately, learning systems are almost all based on very limited abilities to represent information.

- *Planning*

Planning programs start with general facts about the world (especially facts about the effects of actions), facts about the particular situation and a statement of a goal. Then they generate a strategy for achieving the goal. In the most common cases, the strategy is just a sequence of actions.

- *Epistemology*

This is a study of the kinds of knowledge that are required for solving problems in the world.

- *Ontology*

Ontology is the study of the kinds of things that exist. In AI the programs and sentences deal with various kinds of objects, and we study what these kinds are and what their basic properties are. Emphasis on ontology begins in the 1990s.

- *Heuristics*

Heuristics is a way of trying to discover something or an idea imbedded in a program. The term is used variously in AI. Heuristic functions are used in some approaches to search the measure how far a node in a search tree seems to be from a goal. Heuristics predicates that to compare two nodes in a search tree, to see if one is better than the other, i.e. constitutes an advance towards the goal, may be more useful.

What Are the Applications of Artificial Intelligence?

- *Game playing*

You can buy machines that can play master level chess for a few hundred dollars. There is some AI in them, but they play well against people mainly through brute for computation - looking at hundreds of thousands of positions. To beat a world champion by brute and known reliable heuristics force requires being able to look at 200 million positions per second.

- *Speech recognition*

In the 1990s computer speech recognition reached a practical level for limited purposes. While it is possible to instruct some computers using speech, most users have gone back to the keyboard and the mouse as still more convenient.

- *Understanding natural language*

Just getting a sequence of words into a computer is not enough. Parsing sentences is not enough either. The computer has to be provided with an understanding of the domain the text is about, and this is presently possible only for very limited domains.

- *Computer vision*

The world is composed of three-dimensional objects, but the inputs to the human eye and computer TV cameras are two-dimensional. Some useful programs can work solely in two dimensions, but full computer vision requires partial three-dimensional information that is not just a set of two-dimensional views. At present there are only limited ways of representing three-dimensional information directly, and they are not as good as what humans evidently use.

- *Expert systems*

A "knowledge engineer" interviews experts in a certain domain and tries to embody their knowledge in a computer program for carrying out some task. How well this works depends on whether the intellectual mechanisms required for the task are within the present state of AI. When this turned out not to be so, there were many disappointing results. One of the first expert systems was MYCIN in 1974, which diagnosed bacterial infections of the blood and suggested treatments. It did better than medical students or practicing doctors, provided its limitations were observed. Namely, its ontology included bacteria, symptoms, and treatments and did not include patients, doctors, hospitals, death, recovery, and events occurring in time. Its interactions depended on a single patient being considered. Since the experts consulted by the knowledge engineers knew about patients, doctors, death, recover, etc., it is clear that the knowledge engineers forced what the experts told them into a predetermined framework. At the present state of AI this has to be true. The usefulness of current expert systems depends on their users having common sense.

- *Heuristic classification*

One of the most feasible kinds of expert system given the present knowledge of AI is to put some information in one of a fixed set of categories using several sources of information. An example is advising whether to accept a proposed credit card purchase or not. Information about the owner of the credit card, his record of payment and also about the item he is buying and about the establishment from which he is buying it (e.g., about whether there have been previous credit card frauds at this establishment) is available.

What Are the Current Approaches of Artificial Intelligence?

- *A. Knowledge Theory*

To be intelligent requires knowledge and reasoning skills. Intelligent behaviour implies the linking of these two together and, hence, being able to deduce facts that are not explicit in the knowledge and produce sensible reactions to these facts. In humans there is a consciousness that enables us to understand concepts such as what and why, that is intentionality. With this ability we are able to make reasoned judgements and act accordingly. Of course, the "reason" within our decisions is often subjective (and in the same way, our definition of intelligent behaviour is largely subjective). So what forms of reasoning are there? Here are the three main types:

1. Deduction
2. Abduction
3. Induction

The second requirement for intelligent behaviour is the knowledge itself. It is impossible to reason conclusions from knowledge if there is no knowledge. So if we put some facts into a computer system, use a reasoning program into action, we in theory will have an intelligent machine. The reality is that many of these AI structures will work well in simple "toy" domains but once they are presented with real world domain problems and give real world values they suddenly begin to have problems. The problem is that they don't have enough knowledge about the domain and so can't respond to it. If we attempt to simply solve this problem by stuffing more information into the system, we quickly come across the problem of speed. The specific piece of information in the database of knowledge cannot be accessed fast enough for a reasonable response using simple search techniques.

One of the major keys to AI then is being able to store knowledge in an efficient fashion and in such a way that it is possible to compose programs that can access it in a reasonable time. In an ideal world all the knowledge would be incorporated into a system, but this leaves obvious problems. There are no obvious solutions but a number of methods have been proposed that consider knowledge representation like semantic nets, conceptual graphs, frames.

- *B. Expert systems*

An expert system is a class of computer program that can advise, analyze, categorize, communicate, consult, design, diagnose, explain, explore, forecast, form concepts, identify, interpret, justify, learn, manage, monitor, plan, present, retrieve, schedule, test or tutor. They address problems normally thought to require a human specialist for their solution.

Expert based systems are currently in use in business in projects like credit rating people to see if they're worth giving credit to or in the prediction of rise and fall in shares in the stock market. An expert system is based on English, so is easier to be programmed and maintained than in other languages. Expert systems are, however, only experts in their particular field but unlike humans have the advantage of not growing old or making mistakes and can process information faster. The expert based system must have a user interface to gain knowledge and a technique for learning from experience. The system must ask questions and absorb the information given to it. Though there are several problems with this system. The learning requires human intervention, so its knowledge must be given by an expert in the field. Large amounts of memory to hold all the information and a powerful computer to process the data in a reasonable time are required. The other problem is that it can go wrong and this could be expensive for whoever is using the system. Some banks already use these systems in a limited way in the stock market, though there is always human supervision.

Eliza was one of the first expert systems. This is the "computer therapist" program created by an artificial intelligence pioneer Joseph Weizenbaum at MIT. You communicate with this program just as if it were a therapist and it passes the Turing test up to the point. It could be a real person, therefore it is an intelligent program or so you are led to believe. Try and answer these questions: What makes people feel understood by a therapist? What leads to a feeling of rapport? What is it about a conversation that is therapeutic? What is missing in the computer responses that makes it not a real psychotherapist?

- *C. Neural networks*

A neural network is essentially a type of computer but doesn't work in the same way. Conventional computers have a CPU and memory and information is represented in terms of structures of symbols. A neural network however is based upon the structure of the brain which consists of many billions of cells called neurons. These cells are built into a network and electrical signals can pass between neurons at a very fast speed by a conversion into chemical energy. Each neuron in the brain is equivalent to a single processing unit in the computer. The neural network is very similar to passing electrical energy from one point to the other. The mapping out of the basic morphology and of the biological neuron, while being a major step forward, raised the difficult question of how networks of such neurons, that is brain, might carry out the information processing tasks that they obviously perform in people and animals. Some light was cast on this question by McCulloch-Pitts in the early 1940s. These two researchers showed how neuron-like threshold units or TUs might represent logical expressions. From this starting point they went on to show how networks of such units might carry out calculations. In the simplest case a threshold unit is capable of being either on or off. It has several input connections and, in a given time step, receives an input on each connection. The input can be considered to be either a 0 or 1. If the number of 1s received exceeds some threshold the unit outputs a 1, otherwise it outputs a zero. In a network of such units the output on one neuron is the input to others.

McCulloch-Pitts suggested that networks of TUs formed a good model for the function of biological neurons. Such a network can perform computations. Input values can be set on the input units of the network, and a result will be computed on the axons of the output units. The most important feature of neural networks is that they can learn. First they need to be trained. There are various training method and learning rules for neural networks. The problem with neural networks is that they are too expensive to be constructed in anything but the smallest of trials and current computers just lack a sufficient number of pathways between components.

• *D. Fuzzy logic*

Fuzzy logic is a superset of conventional logic that has been extended to handle the concept of partial truth -- truth values between "completely true" and "completely false". Boolean (conventional) logic says that something is either on or off, true or false. You are either sleeping or awake. But what about in-between these times, time between sleeping and a full state of consciousness? When you sit down for a meal, the meal is not just there or not there, there is a continuous period of it being eaten, and each period can be broken down further into more stages of being eaten or not eaten. This idea was introduced by Dr. Lotfi Zadeh in the 1960s as a means to model the uncertainty of natural language.

Zadeh says that rather than regarding fuzzy theory as a single theory, we should regard the process of "fuzzification" as a methodology to generalize *any* specific theory from a crisp (discrete) to a continuous (fuzzy) form. Thus recently researchers have also introduced "fuzzy calculus", "fuzzy differential equations" and so on. Fuzzy logic is used directly in very few applications. The Sony Palmtop apparently uses a fuzzy logic decision tree algorithm to perform handwritten or computer lightpen character recognition. Most applications of fuzzy logic use it as the underlying logic system for fuzzy expert systems. Fuzzy logic and neural networks have been implemented together in recent times and they were even used to control a helicopter with a missing rotor blade. The point is that this could be done quickly enough by a computer but not by a pilot even though initially the program had to be trained by a pilot.

Employment Trends

The demand for AI specialists has changed over the years, reflecting the different stages of development the field has undergone. For years the activity in AI was restricted to college campuses and corporate research centers. Researchers came from a wide variety of academic backgrounds, including biomechanics, computer engineering, computer science, economics, electrical engineering, linguistics, neurobiology, optics engineering, physics, and psychology. Most of them had advanced degrees, usually doctorates. When the technology became commercially viable, many of the scientists involved in research left their laboratories to start their own companies.

At this point, the competition for experienced AI specialists intensified, with salaries increasing significantly as employers competed for available workers. This development, along with the publicity surrounding the success of specific AI products, stimulated more interest in the field. Both the availability of training programs and the number of students taking these courses increased.

Although there are still spot shortages for experienced professionals in certain specialties, the widespread shortage seems to have eased. Salary increases for AI professionals have moderated.

Skill requirements are changing. Requirements are coming down as the field moves out of the research and development phase into product refinement and implementation. Maintaining software based on AI principles is less demanding than developing the original technology. The use of expert system shells has also played a major role in this change. Developing a complex, standalone expert system requires much more programming skill and experience than adapting a shell to the needs of a user does. Because of this evolution, a bachelor's degree in computer science, with an emphasis on AI, is acceptable for a wide array of jobs, although persons involved in the design of AI software still need advanced degrees. Some employers even let people without advanced degrees perform knowledge engineering, generally with shells.

Demand for computer professionals with AI skills will continue to rise. Even if further advances in basic technology aren't made - and this is highly unlikely - the development, integration, implementation, and maintenance of products based on existing technologies will require many additional skilled workers. Demand for workers will also increase as more organizations use AI products.

Growth will occur in software houses and hardware developers producing AI products and in large corporations which are developing their own AI capabilities. In these organizations demand will be the strongest for programmers who can program in LISP or other AI languages. Demand is also growing for knowledge engineers who can work closely between the programmer and the user to design expert systems.

The primary area of growth, however, will be in user organizations. More programmers and systems analysts with AI skills will be needed to integrate shell programs into existing systems and to develop and maintain in-house systems, much as in-house computer professionals develop accounting or database applications today. Artificial intelligence is clearly merging with the field in which most computer professionals are employed, the development and maintenance of management information systems (MIS). In this environment, the strongest demand will be for programmers and systems analysts with experience in regular systems and a good working knowledge of AI.

UNIT II. RELATED TECHNOLOGIES

1. Expert Systems: human expertise in a computer

Beginning in the mid-1960s, a new type of a system, called an *expert system*, began to be developed to support management in the decision-making process. An expert system represents one of the first practical and the most commercially successful applications of artificial intelligence; it is an addition to the kinds of computer systems available to businesses. An expert or knowledge-based system is a computer program that acts as a consultant to decision makers.

The program contains information on a particular subject, known as *the knowledge base* or *the body of facts* (knowledge) and the *heuristics* (thumb rules) that are fed into the computer and are most frequently represented as a series of "if-then" rules, although in some systems the knowledge is represented as frames, objects, or semantic networks. When applied to a problem, the system searches for solutions in the same logical patterns that human experts would use. The largest and most complex expert systems have over a thousand rules and may take 2 years or more to be created. Development takes so long because, for the system to be usable, the expertise to be captured must be clearly defined, and all the steps a human expert would take to draw a conclusion must be spelled out beforehand. This is the job of the knowledge engineer, who is the link between the software developer and the end user.

The expert knowledge and the heuristics are gathered by knowledge engineers — largely through interviews — from human experts in the field for which the computer-based system is being designed to support decisions — fields such as medicine, engineering, or geology. (For example, in the field of medicine, one question that might be asked of an expert system is whether one treatment is better for a patient than another one.) An expert system has the capacity to store the collection of knowledge and manipulate it in response to a user's inquiries; in some cases, it can even explain responses to the user.

Knowledge engineers are somewhat similar to systems analysts. They must not only know the capabilities of the computer hardware and software but also understand the user's operation and what he or she wants to accomplish through automation. They must also be able to deal effectively with people from many different backgrounds - for example, medical researchers, engineers, financial analysts, and industrial machinery mechanics.

The knowledge engineer interviews one or more experts and then distills the information relevant to the application into a set of rules or some other form of knowledge representation that reflects the behaviour of human experts. It can be difficult to get people to part with knowledge they have spent a lifetime acquiring. Even when the experts are eager participants, they can miss the importance of some subtle steps in their decision-making process.

When the knowledge representations are complete, a programmer creates the knowledge base by coding the information in an AI language. In some situations, the knowledge engineer does the actual programming. Finally, there is debugging, testing with trial problems, and refining.

An expert system solves problems that require substantial expertise to understand. An expert system has four major program components:

1. Natural (software) language interface for the user.
2. Knowledge base (like a database, where the facts are stored).
3. Inference machine (software that solves problems and makes logical inferences).
4. Explanation module (which explains its conclusions to the user).

One of the most famous classic expert systems — an older system now being replaced by updated ones — is MYCIN, a system that diagnoses infectious diseases and recommends appropriate medicines. MYCIN represents the knowledge of physicians who are experts in this field, and it enables other, less knowledgeable, physicians to consult with these experts for help in diagnosis and treatment.

For example, bacteremia can be a fatal disease if it is not treated quickly. Unfortunately, traditional tests for it require 24 to 48 hours to verify a diagnosis. However, MYCIN provides physicians with a diagnosis and recommended therapy within minutes. To use MYCIN, the physician enters data on a patient; as the data is being entered, MYCIN asks questions (for example, "Is patient a burn patient?"). As the questions are answered, MYCIN's inference machine "reasons" out a diagnosis: "IF the infection is primary bacteria, AND the site of the culture is the gastrointestinal tract, THEN there is evidence (0.7) that the identity of the organism causing the disease is Bacteroides". The "0.7" means that MYCIN "thinks" there is a 7 out of 10 chance that this diagnosis is correct. This pattern closely follows that of human thought; much of our knowledge is inexact and incomplete, and we often reason using odds (such as "There's a 40 per cent chance it's going to rain") when we don't have access to complete and accurate information.

EMPTY MYCIN, a later development, represented a major advance for AI. It separated the information about blood infections from the inference engine, the rules that apply the knowledge. This breakthrough allowed MYCIN's inference engine to be used with other knowledge bases and led to the development of expert system shells.

An expert system shell is, in effect, an expert system without the expertise. More technically, it is a set of utilities that programmers can integrate into their existing computer system. Users can then fill the shell with the knowledge needed for a specific application. Shell programs permit programmers to create expert systems on desktop computers using conventional languages.

Because shells are much less expensive and more flexible than customized expert systems, their development has caused the overall use of expert systems to skyrocket. No one knows precisely how many expert systems are currently in operation, partly because companies now recognize these systems as competitive weapons and some companies are unwilling to divulge information about them. Nevertheless, knowledgeable sources estimate that several thousands are in use, and the number is growing rapidly. Between one-half and three-fourths of the Fortune 500 corporations now use expert systems; many large corporations have hundreds of systems in use.

Examples of other expert systems are XCON (a system that puts together the best arrangement of Digital Equipment Corporation [DEC] computer system components for a given company), DENDRAL (a system that identifies chemical compounds), PROSPECTOR (a system that evaluates potential geological sites of oil, natural gas, and so on), and DRILLING ADVISOR (a system that assists in diagnosing and resolving oil rig problems).

Capturing human expertise for the computer is a difficult and time-consuming task. "Knowledge engineers" are trained to elicit knowledge (for example, by interview) from experts and build the expert system. The knowledge engineer may program the system in an artificial intelligence programming language, such as LISP or PROLOG, or may use system-building tools that provide a structure. Tools allow faster design but are less flexible than languages. An example of such a tool is EMYCIN, which is MYCIN without any of MYCIN's knowledge. A knowledge engineer can theoretically enter any knowledge (as long as it is describable in rules) into this empty shell and create a new system. The completed new system will solve problems as MYCIN does, but the subject matter in the knowledge base may be completely different.

Expert systems are usually run on large computers — often dedicated artificial intelligence computers — because of these systems' gigantic appetites for memory; however, some scaled-down expert systems (such as the OS/2 version of KBMS, Knowledge Base Management System) run on microcomputers. Scaled-down systems generally do not have all the capabilities of large expert systems, and most have limited reasoning abilities. LISP and PROLOG compilers are available for microcomputers, as are some system-building tools such as EXPERT-EASE, NEXPERT, and VP-Expert, which allow relatively unsophisticated users to build their own expert system. Such expert-system building software tools are called *shells*.

The popularity of expert systems is growing because more and more organizations are finding that they improve quality, raise productivity, cut costs, and increase profits by helping their employees "work smarter". These tools enable computers to deal with ambiguity and questions of judgment that are too subtle for conventional data processing techniques. They also help workers handle much larger quantities of data when making decisions. Regardless of the setting, these systems give the user alternatives ranked according to their probability of success. These probabilities are based on the data available; as more data are added, the reliability of the results improves. Thus, the system appears to learn from the program as it adapts new knowledge into the database.

It also allows users to ask both "What if I do this?" and "What is the best solution for this problem?" Users also can ask the computer the question "Why is this the best solution?" before applying the results. Expert systems help workers at many levels make better decisions by providing them with additional information and a structured way to use that information. These systems are being used in many different types of organizations throughout the economy. For example, public utilities use them to monitor and improve the performance of their coal-fired boilers. About one-fourth of the Nation's largest insurance carriers currently use expert systems, primarily to analyze insurance applications. Manufacturers use them to design products, control processing, and to serve customers who have a problem with a product. Sitting at a computer, employees enter the symptoms and other data provided by the caller, and the program helps them diagnose the problem and prescribe corrective measures.

Expert systems are only one of a cluster of AI technologies with commercial significance. Others involve creating equipment with a human-like ability to move, see, and communicate. The problems encountered so far have driven home to scientists just how complex we humans are and how difficult it is to simulate even our most basic reasoning processes. Getting a robot to do something that a baby does naturally requires hundreds, perhaps thousands, of detailed instructions.

Most researchers now concede that it could be a long time before the necessary breakthroughs occur that will lead to the production of machines that think or reason in any fully human sense of the word, or that act autonomously, or that speak and understand human speech in all its complexity. Nevertheless, researchers keep making advances in these areas.

2. Neural Networks

Another new development is neural network computing, which attempts to duplicate the way the human brain processes information. Conventional computer circuits are arranged in series, with each transistor linked to only two or three others. In neural networks each transistor is hooked up to most, if not all, of the others. A signal entering the system quickly fans out across the entire network and all the transistors process it. This new type of processing can be achieved in two different ways with specially designed neural network computers or with advanced software used on standard computers.

Neural networks are not only faster than conventional computers; they can also develop new if-then rules from the data they receive. They also have superb pattern-recognition capability. For example, once a network has been shown three or four views of a particular face, it will instantly recognize that face from any other angle. This pattern matching has tremendous implications for military applications, product assembly and inspection, and natural language processing.

Neural Systems: an introduction

Humans have always dreamed of creating a portrait of themselves, a ma-machine with humanlike attributes such as locomotion, speech, vision and cognition (memory, learning, thinking, adaptation and intelligence). Through our learning from biological processes and very creative actions, we have been able to realize some of our dreams. In today's technological society we have created machines that have some of the human attributes that emulate several humanlike functions with tremendous capabilities. Some examples of introducing these humanlike functions are human locomotion into transportation systems, human speech and vision into communications systems, and human low-level cognition into computing systems. The subject of intelligent systems today is in such an exciting state of research primarily because of the wealth of information that we researchers are able to extract from the carbon-based computer - the neuronal morphology of the brain, biological sensory systems such as vision, human cognition and decisionmaking processes that form the elements of soft computing.

What Are Neural Networks?

Neural networks (or "nets") are small models inspired by the sorts of small processing units that you find in the human brain (brain cells called "neurons"). Scientists make these models and get them to behave intelligently. They can be taught to do things which traditional computers find difficult. Early neural networks were called Multi-Layer Perceptrons (MLPs).

Another name for neural nets is "connectionist nets", and the people who make and design them are often called "connectionists". The nets often take the form of computer programs that simulate these models, although more and more silicon chips are being produced with neural networks designed directly in the form of electronic components.

The main feature of the nets is that they learn from experience. If you wanted to sort out male faces from the female ones, you would show the net a large number (the more the better) of photographs of faces, and tell it which ones are male and which are female. The net adapts itself so that it learns the differences between male and female faces.

The structure of neurons in the brain was first explained by McCulloch and Pitts **in the 1940s.** *MLPs seemed to be very promising in the early days of computing,* **but they fell out of favour and** only recently they have made a comeback. Nowadays they are starting to appear everywhere, from teaching some small robot to navigate round a maze to acting as an artificial "nose" that can smell the difference between fine wine and plonk. What do neural networks do exactly?

Neural nets are generally used as classifiers. This means that they are used to choose between several options when faced with some input. For instance, it could be given spoken words as input and be required to recognize what the words were (i.e. choosing one word from the possible vocabulary).

They can do other things, of course. Neural networks can act as content addressable memories. This means that they can store complex patterns and then you can prompt them with part of the pattern or a version of the pattern that has been corrupted, and they will retrieve the whole original pattern for you. *A neural network is given a quarter of a pattern and it* reproduces the whole picture from this.

Of course, once you start stitching different neural networks together you can get much more complex behaviour. After all, the human brain is basically that - an enormous number of specialized neural networks all working together to give complex behaviour. Such systems are generally called multinet architectures.

The nets consist of small units called cells, and these are connected to each other in such a way that they can pass signals to each other. In practice, the signal that one cell sends to another is a simple number between 0 and 1, 0 means no signal, 1 means a large signal.

The connections have certain strengths or weights. The net starts off with these connection strengths set randomly. The network is exposed to various inputs and the strengths adjust themselves according to some mathematical plan. This is what we call training *and after it the* network can recognize input patterns or, at least, do something sensible – whatever it has *been trained to do. The information is therefore stored in the strengths of the connections,* **just as it is in the human brain.**

The training is done in tiny steps. With the exception of the WISARD architecture (which trains very quickly), *the training is done as follows:*

1. The first input (training) pattern is presented to the network.

2. The connections are adjusted a tiny amount to improve the network's chances of recog nizing that pattern if it sees it again.

3. The second pattern is presented and step 2 repeated.

4. The same thing happens for all the training patterns.

5. The whole process is rerun with all the training patterns for hundreds (thousands) of times.

The reason the connections are only adjusted a slight amount is to ensure that the networ learns to recognize all the patterns fairly well. If the connections were adjusted a great deal a ter every pattern then the network would be able to recognize the most recently presented pat tern brilliantly and all the others not at all.

Supervised and Unsupervised Training

Most neural nets are trained using supervised training. This means that they are presentec with input patterns and the corresponding desired output patterns, i.e. they are not only tol what the input pattern is but what output, they are supposed to produce when they are givei that input pattern, will be. For instance, a network might be trained to recognize whether a pic ture presented to it was of a man or a woman (like SEXNET by Terry Sejnowski). The networl might be trained to produce a 1 output for a man, a 0 output for a woman (or vice-versa).The training would consist of a large number of pictures presented repeatedly and each accompa nied by a 1 or a 0 as appropriate.

There are still some situations where providing input with corresponding output data (so-called "labelled data") is difficult or inappropriate. For this reason it would be nice if the neural nets could formulate their own decisions as to what the output should be. This is unsupervised training. The networks are given input, but aren't told what they are supposed to produce for output. Instead, they learn to organize it for themselves into sensible regions of their "output space". The standard network for this sort of thing is the Self-Organizing Feature Map (SOFM) by Teuvo Kohonen (Helsinki University). It is set up in such a way that it organizes its connections so that similar input patterns always produce similar patterns of activity in the net. In this way it learns to group similar input patterns together. Of course, it can't put names to those patterns but it does create some sort of order out of the chaos of the different input patterns which are fed into it.

Representing Input

It is important to the success of a neural network that the input signals fed into it are in an appropriate form. Some thought has to be given as to whether the input can be "coded", i.e. transformed from the form in which it appears in the outside world into some more logical form in which patterns may be more obvious.

The Structure of Neural Networks

Neuroscientists tell us that the human brain consists of countless billions of specialized cells called neurons.

In the human brain, neurons are connected in extremely complex networks with countless interconnections. An artificial neural net has a much simpler structure. The most common structure is called a feedforward Multi-Layer Perception where neurons are arranged in layers, with the outputs of neurons in each layer being connected to the inputs of the neurons in the layer above.

Usually, the net has three layers called the Input layer, the Hidden layer and the Output layer. The input to the entire net is presented in the input layer and is fed up into the layers, with the output layer providing the output from the neural net that the user requires.

Each connection from one node to another carries a strength which indicates how important the connection is. Strong connections have more influence on the node they connect into than weaker ones. They contribute more to the firing of the cell. The information carried by the network is stored in the differing strengths of the node connections.

Nowadays more and more neural networks are being implemented in specially created integrated circuits. However, most programmers simulate neural networks using software. The strengths between the nodes are called weights in the program and are stored as numbers.

The term "feedforward" means that the connections between one layer and the next only run in one direction. There are connections from layer 1 to layer 2, from layer 2 to layer 3 etc. but no connections in the other direction. The opposite of a feedforward net is called a recurrent net, which have feedback connections.

Recurrent networks often have attractor states. This means that signals passing through the recurrent net are fed back and changed until they fall into a repeating pattern, which is then stable (i.e. it repeats itself indefinitely as it rattles round the loop). This is a little like a ball being placed on a slope and released - it rolls downhill until it reaches the bottom of a valley, and then stops. The input signals change until they reach one of these attractor states, and then they remain stable. The secret with recurrent networks is to train the weights so that the attractor states are the ones that you want. The sort of problems that neural networks are asked to sort out can also be classified as linearly separable or linearly inseparable. Linearly separable tasks are a lot easier to teach to neural networks than linearly inseparable ones. Indeed, there are certain types of network (simple perceptrons, for example) which simply can't learn linearly inseparable tasks.

3. Robotics

There's no precise definition of a robot. It is normally defined as a programmable machine imitating an intelligent creature. Getting information from its surroundings and doing something physical (moving or manipulating objects) qualify a machine as a robot.

According to Webster's Ninth New Collegiate Dictionary, a robot is an automatic device that performs functions ordinarily ascribed to human beings or that operates with what appears to be almost human intelligence. In the field of artificial intelligence, there are intelligent robots (also called *perception robots)* and unintelligent robots. Most robots are unintelligent; that is, they are programmed to do specific tasks, and they are incapable of showing initiative. An unintelligent robot cannot respond to a situation for which it has not been specifically programmed. Intelligence is provided either by a direct link to a computer or by on-board computers that reside in a robot. Robotic intelligence is primarily a question of extending the sensory (for example, vision) and mobility competence of robots in a working environment.

Robots are increasingly useful. Industrial robots have been around for about 25 years, doing simple, repetitive tasks requiring no decision-making, and doing them with superhuman speed and precision. They now perform a remarkably wide variety of tasks, from assembling computers, artillery shells, and vacuum cleaners to popping frozen dinners into their trays, inspecting different kinds of products, and drilling holes for brain surgery. They are also used in hazardous environments, such as mines and nuclear power plants, and for handling toxic waste.

In the future, reasoning ability will be incorporated into robots, thus improving their ability to behave "intelligently". Robot vision has already been successfully implemented in many manufacturing systems. To "see", a computer measures varying intensities of light of a shape; each intensity has a numerical value that is compared to a template of intensity patterns stored in memory. One of the main reasons for the importance of vision is that production-line robots must be able to discriminate among parts. General Electric, for example, has Bin Vision Systems, which allow a robot to identify and pick up specific parts in an assembly-line format.

Another area of interest is the "personal" robot, familiar to us from science fiction. Existing personal robots exhibit relatively limited abilities, and whether a sophisticated home robot can be made cost-effective is debatable. B.O.B. (Brains On Board) is a device sold by Visual Machines that can speak (using prerecorded phrases), follow people around using infrared sensors, and avoid obstacles using ultrasonic sound. Software will allow the robot to bring its owner something to drink from the refrigerator. The performance limitations of personal robots reflect the difficulties in designing and programming intelligent robots. In fact, we have just begun to appreciate how complicated such mundane tasks as recognizing a can of Pepsi in the refrigerator can be.

Significant improvements have been made in robotics technology in recent years. For example, to reprogram a new robot arm, an operator simply selects from among choices that the robot itself offers; a worker can teach the robot's eyes to recognize a new part in less than 10 minutes. The number of robots in use has grown to over 32,000. As prices fall, they will become more attractive to potential users, especially as employers continue to experience difficulty attracting workers. Robot technology may be especially desirable in fast-food preparation, some areas within health care, building maintenance, and security, industries that traditionally have relied on young people.

Name a boring or dangerous job. Somewhere, a robot is probably doing it. Robots are ideal for doing jobs that require repetitive, precise and fast movements. Robots are good at doing the same thing; without asking for a safe working environment, salary, breaks, food and sleep, without getting bored or tired, without making mistakes. Factories are so highly automated that most human workers carry out only supervising and maintaining the robots.

People keep finding new uses for robots - making and packing drugs and food, soldering tiny wires to semiconductor chips inserting integrated circuits onto printed circuit boards used in electronics, working in radioactive "hot zones", exploring space.

All work and no play make anyone dull - even a robot. Soccer-playing robots gather each year at RoboCup, an international event collecting over 100 teams from 35 countries.

Robotic players use radio signals to coordinate their actions with their teammates.

Teams are placed in divisions based on size, ranging from the size of a pizza box. By 2050, the organizers of RoboCup count on developing a team of fully autonomous humanoid robots that can beat the human world champion team in soccer.

The History of Robotics

If you think robots belong to space movies, think again. Now robots are all over the world. Putting chocolates into boxes, walking into live volcanoes, driving trains in Paris and defusing bombs in Northern Ireland are their common tasks. Today's robots are doing more and more things humans can't do or don't want to do.

The idea of creating an intelligent machine is very old. Homer described gold girls, mechanical helpers built by Hephaistos, the Greek god of smiths. Aristotle is credited with the original concept of automation. In the fourth century B.C., he wrote "If every instrument could accomplish its own work, obeying or anticipating the will of others...if the shuttle could weave, and the pick touch the lyre, without a hand to guide them, chief workmen would not need servants...". In 1495, Leonardo da Vinci designed a mechanical man. But only the invention of transistors and integrated circuits in the 1950s and 1960s made real robots possible. Compact, reliable electronics and computers added brains to already existing machines. In 1959 researchers demonstrated the possibility of robot manufacturing ash-trays.

A Czechoslovakian author named Karel Chapek introduced the word "robot" to the public in his 1921 play R.U.R., "Rossum's Universal Robots" where robots are invented to help people by performing simple tasks, but being used to fight wars they turn on their human masters and take over the world. The Czech word "robota" means "hard work", thus the Czech robot is defined as "a worker of forced labour".

Similarly, the Random House College Dictionary defines the robot as "a machine that resembles a man and does mechanical, routine tasks on command". In his "Robots: Machines in Man's Image" Asimov states that industrial robots do not have the "human form appearance of their robotic brethren". According to the Robotic Industries Association, an industrial robot is a reprogrammable, multifunctional manipulator designed to move materials, parts, tools or specialized devices through variable programmed motions to perform a variety of tasks.

The first generation of robots consisted of *automatons*. An automaton is a self-moving machine, constructed for the purpose of imitating the motions of men or animals (Asimov). Most of the earliest automatons were clock controlled ornamentations. In 1350 an automated rooster was erected on top of the Cathedral in Strasbourg. Every day at noon, it would flap its wings and crow. In 1497 two bell striking giants were built on top of the great clock tower in Piazza San Marco, Venice. Within the same time period, an Arab named al-Jazari wrote a book on automatons. The book included an illustration of an automated Arab lady that filled and emptied a wash basin. The modern toilet was derived from the same principle.

Later versions of automatons were based on self-contained clockwork mechanisms. In 1738 Jacques de Vaucanson invented an entertaining mechanical duck that quacked, flapped its wings, drank water, ate food and discarded waste. In 1774 Droz invented one of the most complicated automatons in history. The "automatic scribe" could write any message up to 40 characters long. In 1805 Maillardet built a spring-activated automaton that could draw pictures and write in both French and English. At the 1876 World's Fair life sized automatons including brass instrument players, artists, and card magicians entertained large audiences. Within a few years Thomas Edison used a condensed version of his phonograph invention in the design of the famous talking doll.

After Chapeks R.U.R. *electromechanical automatons* were referred to as *robots*. In 1928 an electromechanical robot was built in London. Although the robot contained an electric motor, electromagnets, pulleys, and wheels, it could not function beyond its platform. In 1940 Westinghouse created two of the first robots that used the electric motor for entire body motion in the rectangular coordinate plane. "Electro" danced, counted to ten, smoked, and announced the latest Westingouse products. His motorized companion dog walked, stood on its hind legs and barked.

After the invention of the transistor in 1948 many robots were used in conjunction with the computer. The first patent for *a computer controlled industrial robot* was developed in 1954 by George Devol. Devol created a computerized memory and control system called "universal automation". Devol co-founded the Unimation industrial robot company and "started the industrial robot revolution" by selling designs of powerful assembly line arms to General Motors. In the late 1960s researchers developed a computer controlled robot called Shakey. Besides moving between rooms and avoiding objects, Shakey II was able to stack wooden blocks according to spoken instructions. It watched if the blocks were properly aligned, and if not, it adjusted the stack. Shakey was once asked to push a box off a platform, but it could not reach the box. The robot found a ramp, pushed the ramp against the platform and then pushed the box onto the floor.

Hughes Aircraft created the Mobots in the late 1960s. Mobots were controlled by remote radio and camera systems operated by either people or computers. They were designed for "environments beyond [human] capacity and for tasks beyond [human] capability". Applications included construction, chemical testing, and nuclear reactor environment interaction.

The parts of many modern robots can be generalized into four categories: the base, object manipulator, primary control system and sensory system. *The base* is usually a metal or a plastic frame that supports the robot's components. Most industrial robot bases are stationary, although the arms move about. Other bases move about by treads, wheels or legs. Wheel driven bases have various configurations. Some have two big rear wheels, and a small front balancing wheel, while others have four equally sized wheels. An interesting example of a walking base is the six-legged ODEX I. Bases with hopping and galloping legs have also been developed.

The second part of the modern robot is *the object manipulator*. Basic grasping and manipulation requires a large amount of memory due to the requirements of smoothness and sensitivity during operation. The minimum number of fingers necessary to grasp an object, hold it securely and manipulate it smoothly was found to be three. The Best of National Geographic film featured a three-fingered robot that flawlessly manipulated objects including a screw driver, a coke can and a pen. Current household robots have simple gripping mechanisms. Common industrial robot object manipulators include a box gripper, a cylinder gripper, a suction cup, a ladle, a spotwelding gun, a drill, a torch and a grinder.

The third part of the modern robot is *the control system.* Primary systems include the remote control, the driver circuit or the computer. Quite often the control system consists of a primary control and secondary application-specific controls. The primary control executes the main program, calling individual functions or reading resultant data, while the secondary control systems determine how those functions are processed. For example, ODEX I has a microprocessor-based motion control system. Each leg has a sensor that sends messages to a corresponding microprocessor. A total of six microprocessors are directed by a "central motion control" microprocessor. Dedicating individual microcontrollers to the motion control of their corresponding legs is now common practice by numerous organizations (Texas Tech group).

The final part of the modern robot is *the sensory system.* The sense of touch is used for object recognition or collision avoidance. For example, a robot hand, equipped with a rubber skin of microswitches, can recognize objects such as screws, pins and washers. Examples of tactile sensors used for avoiding obstacles include metallic loop "feelers" and bumpers with microswitches or conductive foam / backplate contacts. The sense of hearing enables a robot to react to spoken commands and sounds within a specified frequency. The sense of sight enables a robot to recognize colours, shapes and patterns. The CYBER I robot is equipped with an "ultrasonic vision system" that is used for avoiding obstacles.

Robots rely on sensors to get information about their surroundings. In general, a sensor measures an aspect of the environment and produces a proportional electric signal. Many of a robot's sensors mimic aspects of our own senses, but not all of them. Robots can also sense those things we can't like magnetic fields or ultrasonic sound waves.

Robotic light sensors come in many different forms – photoresistors, photodiodes, phototransistors – but they all have roughly the same result. When light falls on them, they respond by creating or modifying an electric signal. A filter put in front of a light sensor can be used to create a selective response, so the robot only "sees" a certain colour.

Light sensors can also be used for simple navigation, for example, by allowing a robot to follow a white line. Other robots navigate using infrared light (the same invisible light used in your TV remote control). A robot sends out a beam of infrared light, some of which bounces off of an obstacle and returns to a light sensor on the robot.

For more elaborate vision systems simple light sensors are not enough. Robots like the ones that find and remove imperfect products from a conveyor belt need to be able to resolve complex, changing images quickly. In these situations the image from a camera "eye" must be broken down and analyzed by a computer program.

Robotic vision has proved to be one of the greatest challenges for engineers. Difficulty lies in programming a robot to see what's important and to ignore what isn't; a robot has trouble interpreting things like glare, lighting changes and shadows. Also for a robot to have depth perception, it needs stereoscopic vision like our own. Resolving two slightly different images to make one 3-D image can be a computational nightmare, requiring large amounts of computer memory.

A good example of a household applications robot is the well known Arok, which has been featured in books and on television. The voice activated robot can perform 36 functions including vacuuming, walking the dog and lifting heavy objects. An older model of a Heathkit robot (HERO I) has a sound sensory system, an ultrasonic ranging system, a visible light spectrum analyzer, a motion detector, a speech synthesizer, a clock, a gripper arm and a 6808 microprocessor. It can be programmed to perform simple tasks like playing games or highly sophisticated tasks, such as guarding a house. Robots can also be used to help the disabled. The Palo Alto Veteran's Administration Center developed a voice controlled helping hand (Ferrel). The helper could prepare dinner, serve drinks, retrieve desk files, draw, and turn book pages.

In the fourteenth century clockwork automatons were created for decoration purposes. Eighteenth century automatons were used for entertainment purposes. In the early twentieth century inventors used electromechanics to modify the automaton into the robot. The transistor was used in the development of computer controlled experimental and industrial robots. The latest robotic applications were made possible through the use of microprocessors. With further innovations the robot will undoubtedly become a daily necessity.

Advances in Robotics

A robot is a machine that gathers information about its environment (senses) and uses that information (thinks) to follow instructions to do work (acts).

Imitating humans, robots also sense magnetic fields and ultrasonic waves. Robotic light sensors work by creating or changing an electric signal when light falls on them. When navigating, the robot sends out a beam of infrared light which bounces off objects and returns to a light sensor of the robot. However, making 3D images requires large amounts of computer memory.

The ability to move sets robots apart from computers. A mechanical device for producing motion is known as an actuator. A single robot is supplied with dozens of actuators, each chosen to do a specific task. Electric motors are actuators that produce motion from electricity by the electromagnetic effect. Their high speed and a small turning power make a gearbox necessary. Special stepper motors turning in precise "steps" are ideal for adjusting position. A servomotor is used for turning only 90° to the right or left. If you've ever driven a toy car, boat, or plane by remote control, a servomotor was probably responsible for the steering. Solenoids are electric motors for producing linear or in-and-out motion. Solenoids are used in switches turning things off and on. Although making a robot move like a person is not easy, engineers at Honda have designed robots capable of walking, climbing stairs and keeping their balance - no two-legged robot has ever done it before.

How to make robots think? There are three approaches to artificial intelligence.

Most robots have a microcomputer for "brains", which allows programming a lot of information. But they work only according to their programme and cannot learn. Neural networks are modelled after the human brain. A neural net "learns" by exposure to lots of input aid corresponding output. Once trained, the neural net responds to an input with a likely output. Unlike rule-based systems, neural networks are incapable of giving definite answers. Stimulus-response robots pioneered by Rodney Brooks at MIT have no memory and no logical decision-making - only hard-wired responses to stimulation.

Can a robot be conscious? Can it be not only intelligent, but aware in the way that we are? So far artificial intelligence has never shown such signs of life. However, if robots eventually think like us, detect and express emotions, pursue their own interests (whatever those are programmed to be) and even make copies of themselves, drawing a line between machines and living things will be increasingly difficult.

In 1950 the British mathematician Alan Turing proposed a test to answer the question, "Can machines think?" The Turing test goes something like this:

A human interrogator can ask any question of two subjects, which are both in other rooms. One of these subjects is another person, the other one is a machine such as a computer – the interrogator must decide which is which. The answers are typewritten, so there are no clues from voice or appearance. What's more, the subjects don't have to tell the truth, so the computer can pretend to be human. If the human interrogator can't tell the difference between the person and the machine, then the answer to the question «Can machines think?» has to be «yes».

Some scientists see silicon-based life forms as the next step in evolution, replacing carbon-based life forms like us. Talk of robots taking over sounds a little strange, yet many respectable scientists (Hans Moravec, Ray Kurzweil, Bill Joy) think it is likely that robots will play a growing and even a dominant role in the future. Imagine: robots become so intelligent and so capable that we come to rely on them in everything. Useless and unnecessary, humans are gradually pushed aside.

Some scientists predict that advances in robotics, genetic engineering and nanotechnology will lead to a world populated by super organisms both biological and mechanical. When we build machines that are like us, only smarter, stronger and more easily produced, they say, we are in fact creating our own worst enemy. These machines will develop new forms of thinking that will be beyond our comprehension. If we can't understand what we have built, we will not be able to control it. The scientists say that we have some 20 years of intellectual superiority over computers. By that time the robots will have learnt a lot. They will deserve the same rights and privileges because they will be like humans.

We're probably decades away from having to worry about anything more than running out of batteries. Still it seems clear that big changes are coming, and as with any new technology, there will certainly be some unintended and quite possibly unpleasant consequences as robots begin to play a regular role in our day-to-day lives. But the potential benefits outweigh the risks. Let's hope that the society is strong and wise enough to stop abuses without stopping science.

Fear of machines wising up and taking control is not new. Likewise, there's no shortage of movies that depict super-intelligent robots that turn on their human creators: *The Matrix*, *Terminator*, and *2001: A Space Odyssey* are a few.

Scientist-turned-writer Isaac Asimov wrote many science fiction tales that featured robots as characters. In Asimov's stories, the robots were guided by a set of rules, called "The Three Laws of Robotics" which prevented robots from harming people. They are:

1. A robot may not injure a human being or, through inaction, allow a human being to come to harm.

2. A robot must obey the orders given to it by human beings, except when such orders would conflict with the first law.

3. A robot must protect its own existence, as long as this does not conflict with the first two laws.

Although Asimov wrote these laws as fiction in the 1940s, before robots existed, they reflect ongoing concerns that some people have about robots. Technically, destructive technologies like "smart" cruise missiles (which can be considered robots) are already violating Asimov's laws.

However, if robots do develop consciousness, they may also develop conscience, and choose to be kind to their human creators. In the meantime, we may want to remember where the "off" button is. . .just in case.

4. Artificial Reality

Want to take a trip to the moon? Be a racing car driver? See the world through the eyes of an ocean-bottom creature or your cat? Soon, without leaving your chair, you will be able to experience almost anything you want through the form of AI called *artificial reality* (Figure 1), also known as *virtual reality* and *virtual environments*. In virtual reality, the user experiences a computer-generated environment called *cyberspace;* he or she is "inside" a world instead of just observing an image on the screen. To put yourself into artificial reality, you need special hardware — a headset called *Eyephones* with 3-D screens and earphones, and gloves called *DataGloves*, which collect data about your hand movements and recognize commands from hand gestures.

The headset includes a head-tracking device to enable the viewpoint to change as you move your head. The hardware uses software, such as Body Electric, that translates data into images and sound. Aside from entertainment, artificial reality can provide instructional simulation situations to help people learn to exercise skills under varying conditions — skills such as surgery, driving, flying, outerspace operations, police work, and disaster management, to name but a few.

How 3-D PC Glasses Work

Only a few years ago, seeing in 3-D meant peering through a pair of red-and-blue glasses, or trying not to go cross-eyed in front of a page of fuzzy dots. It was great at the time, but 3-D technology has moved on. Scientists know more about how our vision works than ever before, and our computers are more powerful than ever before - most of us have sophisticated components in our computer that are dedicated to producing realistic graphics. Put those two things together, and you'll see how 3-D graphics have really begun to take off.

Most computer users are familiar with 3-D games. Back in the '90s, computer enthusiasts were stunned by the game Castle Wolfenstein 3D, which took place in a maze-like castle. It may have been constructed from blocky tiles, but the castle existed in three dimensions - you could move forward and backward, or hold down the appropriate key and see your viewpoint spin through 360 degrees. Back then, it was revolutionary and quite amazing. Nowadays, gamers enjoy ever more complicated graphics - smooth, three-dimensional environments complete with realistic lighting and complex simulations of real-life physics grace our screens.

But that's the problem - the screen. The game itself may be in three dimensions, and the player may be able to look wherever he wants with complete freedom, but at the end of the day the picture is displayed on a computer monitor... and that's a flat surface.

That's where PC 3-D glasses come in. They're designed to convince your brain that your monitor is showing a real, three-dimensional object. In order to understand quite how this works, we need to know what sort of work our brain does with the information our eyes give it. Once we know about that, we'll be able to understand just how 3-D glasses do their job.

Seeing in Three Dimensions

Human beings, like most other creatures, are equipped with two eyes, situated close together and side by side. This positioning means that each eye has a view of the same area from a slightly different angle. You can check this out by focusing on a distant object and viewing through each eye alternately and see how some things seem to change position slightly.

The brain takes the information from each eye and unites them into one picture, interpreting the slight differences between each view as depth. This produces a three-dimensional picture: one with height, width and depth.

It is the added perception of depth that makes 3-D or stereoscopic vision so important. With stereoscopic vision we see exactly where our surroundings are in relation to our own bodies, usually with considerable precision. We are particularly good at spotting objects that are moving towards or away from us, and the positioning of our eyes means we can see partially around solid objects without needing to move our heads. It's easy to see why some people believe stereoscopic vision to have evolved as a means of survival.

Certainly, stereoscopic vision is vital for seemingly simple actions such as throwing, catching or hitting a ball, driving or parking a car, or even just threading a needle. That's not to say such tasks can't be managed without 3-D vision, but a lack of depth perception can make these everyday tasks much more complex.

A Different Point of View

The key to stereoscopic vision is depth, and our brain will happily take care of that for us, providing our eyes are given the right information in the first place. This is exactly how those red-and-blue glasses work - each colour filters out part of the image, giving each eye a slightly different view. The brain puts the two different images together, and those blue-and-red blurry images are turned into a fantastic 3-D comic, or movie, or TV show.

Stereograms, also known as Magic Eye pictures, use seemingly-random patterns of dots but rely on the viewer to cross his eyes in just the right way or to look through the image until the eyes see just the right part and allow the brain to decode the hidden depth information.

Both methods have their disadvantages, of course - the red-and-blue glasses make it difficult to show colour in the 3-D image, and viewing stereograms is an art in itself. Neither method is entirely suitable for playing games.

Nevertheless, the underlying principle is exactly the same: creating and controlling those two different points of view. But just how easy is it to create these two separate images, one for each eye?

The answer is all about how games are created. Not so long ago the graphics we saw on our computer screens was carefully drawn into the computer - every single frame of animation, every different view of a character. If you wanted a dinosaur in your game, you sat down and drew the different views of a dinosaur into the computer.

Nowadays, game designers sit down with a 3-D graphics package and design their dinosaur in three dimensions. The computer has a 3-D model of the dinosaur in its memory, and the game simply works out where the player is looking and draws the correct view of the dinosaur using the 3-D model. In fact, everything you see on your screen in a modern 3-D game is produced in the same way; the game is like a gigantic 3-D model. The computer works out what it needs to display on your screen and generates the appropriate view.

Since the computer is able to create one point of view, there's no problem shifting the viewpoint slightly and creating another point of view. And after that, all you need is a way to get the correct image to the correct eye.

Getting Synched

It's all down to the power of liquid crystal displays, or LCD. Just like the liquid crystal in a watch can be changed from transparent to black, the lenses of PC 3-D glasses can be transparent or opaque. In other words, the glasses can control which eye sees the image on the screen, and with careful timing you've got perfect 3-D. Here's how it happens:

1. The images are prepared by the computer and displayed. Two images are generated, representing the views seen by each eye.

Both of these views are presented on the screen in rapid sequence.

2. While the left view is presented, the right eye is blocked by the LCD glasses. Similarly, when the right view is presented, the left eye is blocked.

All of this happens so quickly that the brain is entirely unaware of the two images merging together into a stereoscopic view. This is the same thing as when we watch a film using an old film projector and the sequence of still images flickering onto the screen merges together to form a movie.

A Brief History of 3-D Glasses

So, we've seen that although there might be something complex going on behind the scenes, with the right equipment we can just sit back and let our eyes do the work. Of course, the technology wasn't always so simple; there have, in fact, been four generations leading up to today's 3-D glasses.

The first generation modified the games themselves to make them compatible with stereoscopic 3-D. The game creators had to specifically support each type of LCD glasses – hardly an ideal situation. There was no guarantee that the glasses you'd bought would work with your favourite game. As you can imagine, that didn't appeal to many people; so one more solution was developed.

This second solution was to override the game, actually taking over the computer screen and altering what was displayed. As far as the game was concerned, it was just doing what it normally did, except, of course, that some of the computer's time was taken up processing the image to make it 3-D. The result was slower performance and low-resolution, blocky images. It did work with hundreds of games, though, it was a definite improvement.

The third generation worked in a similar way, modifying the graphics driver, but also maintaining the resolution of the images. Unfortunately, it wasn't compatible with many games, though it was a definite forerunner to the 3-D glasses we have nowadays.

In the fourth-generation models, compatibility is high, the complicated work is done by the graphics card, and the lightweight LCD glasses flick so rapidly between the two images that all we see is crystal-clear, 3-D images.

5. Pattern Recognition

Image Processing Technologies

Pattern Recognition

Pattern recognition is a branch of machine learning. It can be defined as the act of taking in raw data and taking an action based on the category of the data.

Pattern recognition is an important field of computer science concerned with recognizing patterns, particularly visual and sound patterns. It is central in optical recognition, voice recognition and handwriting recognition.

It's a process of identifying a stimulus, recognizing a correspondence between a stimulus and information in permanent (LTS) memory. This process is often accomplished with incomplete or ambiguous information. Many variations on a pattern may be recognized as the same object or a class of objects.

Most research in pattern recognition is about methods for supervised learning and unsupervised learning.

Pattern recognition aims to classify data (patterns) based on either *a priori* knowledge or on statistical information extracted from the patterns. The patterns to be classified are usually groups of measurements or observations, defining points in an appropriate multidimensional space.

A complete pattern recognition system consists of a sensor that gathers the observations to be classified or described; a feature extraction mechanism that computes numeric or symbolic information from the observations; and a classification or a description scheme that does the actual job of classifying or describing observations, relying on the extracted features.

The classification or description scheme is usually based on the availability of a set of patterns that have already been classified or described. This set of patterns is termed the training set and the resulting learning strategy is characterized as supervised learning. Learning can also be unsupervised, in the sense that the system is not given *a priori* labelling of patterns, instead it establishes the classes itself based on the statistical regularities of the patterns.

The classification or description scheme usually uses one of the following approaches: statistical (or decision theoretic), syntactic (or structural). Statistical pattern recognition is based on statistical characterizations of patterns, assuming that the patterns are generated by a probabilistic system. Structural pattern recognition is based on the structural interrelationships of features. A wide range of algorithms can be applied for pattern recognition, from very simple Bayesian classifiers to much more powerful neural networks.

Holographic associative memory is another type of pattern matching scheme where target small patterns can be searched from a large set of learned patterns based on cognitive meta-weight.

Typical applications are automatic speech recognition, classification of text into several categories (e.g. spam/non-spam e-mail messages), the automatic recognition of handwritten postal codes on postal envelopes, or the automatic recognition of images of human faces. The last two examples form the subtopic image analysis of pattern recognition that deals with digital images as input to pattern recognition systems.

Pattern recognition is studied in many fields, including psychology, ethnology, and computer science.

Image Processing Technologies

The long-term goal of most advanced computer vendors is to create systems that are capable of handling compound documents that integrate data from different applications - text, data, spreadsheet, graphic, image, and voice information - into a cohesive information system. Compound document systems conceivably allow an architect, for example, to:

1. Turn on the system and orally ask it to display sketches of a new amusement park, which are stored in the database
2. Combine these sketches with photos of three proposed sites
3. Attach a written purchase order
4. Add a short oral note asking for comments
5. Send the whole package electronically across the country to the client's office

While such futuristic document systems will not appear overnight, some compound technologies, such as word processing and spreadsheets, are well established. Other technologies such as image and voice processing may not mature for many years.

Beginning computer users are often surprised by how difficult it is to combine drawings, graphics and photographs into documents. The technical problems arise from the huge amount of storage space required to store graphics; for example, while one page of a text requires about two kilobytes of storage, an 8 x 10-inch photograph can require several megabytes—enough computer space to hold two or three thousand pages of a text. Image processing systems that allow users to manipulate images in databases and combine them with a text are slowly maturing, however, and will become more common throughout this decade.

Current image systems often employ workstations and laser printers; in addition, they may communicate with other devices via local and wide area networks. They often contain specialized devices such as *scanners* (devices that convert pictures and text into machine-readable information), *optical disks* (platters about the size of long-playing records that can contain about 40,000 pages of images and a text), and *jukeboxes* that manipulate optical disks to process images.

A good image processing system also contains software that allows users to index, compress, edit, annotate, cut, paste, and route image information. In addition, image processing systems often contain word processors that allow users to combine texts and graphics.

For image systems to work effectively in organizations, three technological hurdles must be overcome: data transfer rates must increase, users must be able to write to optical disks, and software tools capable of integrating images with other office information need to be developed.

6. Facial Recognition

A *facial recognition system* is a computer-driven application for automatically identifying a person from a digital image. It does that by comparing selected facial features in the live image and a facial database.

It is typically used for security systems and can be compared to other biometrics such as fingerprint or eye iris recognition systems.

Popular recognition algorithms include eigenface, fisherface, the Hidden Markov model, and the neuronal motivated Dynamic Link Matching. A newly emerging trend, claimed to achieve previously unseen accuracies, is three-dimensional face recognition. Another emerging trend uses the visual details of the skin, as captured in standard digital or scanned images. Tests on the FERET database, the widely used industry benchmark, showed that this approach is substantially more reliable than previous algorithms.

In the 2000 presidential election, the Mexican government employed facial recognition software to prevent voter fraud. Some individuals had been registering to vote under several different names, in an attempt to place multiple votes. By comparing new facial images to those already in the voter database, authorities were able to reduce duplicate registrations. Similar technologies are being used in the United States to prevent people from obtaining fake identification cards and driver's licenses.

There are also a number of potential uses for facial recognition that are currently being developed. For example, the technology could be used as a security measure at ATM's; instead of using a bank card or personal identification number, the ATM would capture an image of your face, and compare it to your photo in the bank database to confirm your identity. This same concept could also be applied to computers; by using a webcam to capture a digital image of yourself, your face could replace your password as a means to log-in.

Despite the potential benefits of this technology, many citizens are concerned that their privacy will be invaded. Some fear that it could lead to a "total surveillance society" with the government and other authorities having the ability to know where you are, and what you are doing, at all times.

Early Development

During 1964 and 1965, Bledsoe, along with Helen Chan Wolf and Charles Bisson, worked on using the computer to recognize human faces. He was proud of this work, but because the funding was provided by an unnamed intelligence agency that did not allow much publicity, little of the work was published. Given a large database of images and a photograph, the problem was to select from the database a small set of records such that one of the image records matched the photograph. The success of the method could be measured in terms of the ratio of the answer list to the number of records in the database. Bledsoe described the following difficulties.

This recognition problem is difficult because of the great variability in head rotation and tilt, lighting intensity and angle, facial expression, aging, etc. Some other attempts at facial recognition by machine have allowed for little or no variability in these quantities. Yet the method of correlation (or pattern matching) of unprocessed optical data, which is often used by some researchers, is certain to fail in cases where the variability is great. In particular, the correlation is very low between two pictures of the same person with two different head rotations.

This project was labelled man-machine because the human extracted the coordinates of a set of features from the photographs, which were then used by the computer for recognition. Using a graphics tablet (GRAFACON or RAND TABLET), the operator would extract the coordinates of features such as the center of pupils, the inside corner of eyes, the outside corner of eyes, point of widow's peak, and so on. From these coordinates, a list of 20 distances, such as width of mouth and width of eyes, pupil to pupil, were computed. These operators could process about 40 pictures an hour. When building the database, the name of the person in the photograph was associated with the list of computed distances and stored in the computer. In the recognition phase, the set of distances was compared with the corresponding distance for each photograph, yielding a distance between the photograph and the database record. The closest records are returned.

This brief description is an oversimplification that fails in general because it is unlikely that any two pictures would match in head rotation, lean, tilt, and scale (distance from the camera). Thus, each set of distances is normalized to represent the face in a frontal orientation. To accomplish this normalization, the program first tries to determine the tilt, the lean, and the rotation. Then, using these angles, the computer undoes the effect of these transformations on the computed distances. To compute these angles, the computer must know the three-dimensional geometry of the head. Because the actual heads were unavailable, Bledsoe used a standard head derived from measurements on seven heads.

After Bledsoe this work was continued at the Stanford Research Institute, primarily by Peter Hart. In experiments performed on a database of over 2000 photographs, the computer consistently outperformed humans when presented with the same recognition tasks.

Among the different biometric techniques facial recognition may not be the most reliable and efficient but its great advantage is that it does not require aid from the test subject. Properly designed systems installed in airports, multiplexes, and other public places can detect presence of criminals among the crowd. Other biometrics like fingerprints, iris, and speech recognition cannot perform this kind of mass scanning. However, questions have been raised on the effectiveness of facial recognition software in cases of railway and airport security.

7. Handwriting Recognition

The technique by which a computer system can recognize characters and other symbols written by hand is natural handwriting. The technology is used for identification and also on devices such as PDA and tablet PCs where a stylus is used to handwrite on a screen, and then the computer turns the handwriting into a digital text.

Handwriting recognition is the ability of a computer to receive intelligible handwritten input. The image of the written text may be sensed "off line" from a piece of paper by optical scanning (optical character recognition). Alternatively, the movements of the pen tip may be sensed "on line", for example by a pen-based computer screen surface.

Handwriting recognition principally entails optical character recognition. However, a complete handwriting recognition system also handles formatting, performs correct segmentation into characters and finds the most plausible words.

On-line recognition

On-line handwriting recognition involves the automatic conversion of texts as they are written on a special digitizer or PDA, where a sensor picks up the pen-tip movements X(t),Y(t) as well as pen-up/pen-down switching. That kind of data is known as digital ink and can be regarded as a dynamic representation of handwriting. The obtained signal is converted into letter codes which are usable within computer and text-processing applications.

The elements of an on-line handwriting recognition interface typically include:

- a pen or stylus for the user to write with.
- a touch sensitive surface, which may be integrated with, or adjacent to, an output display.
- a software application which interprets the movements of the stylus across the writing surface, translating the resulting curves into digital texts.

Handwriting recognition is commonly used as an input method for PDAs. The first PDA to provide written input was the Apple Newton, which exposed the public to the advantage of a streamlined user interface. However, the device was not a commercial success, owing to the unreliability of the software, which tried to learn a user's writing patterns. By the time of the release of the Newton OS 2.0, wherein the handwriting recognition was greatly improved, including unique features still not found in current recognition systems such as modeless error correction, the largely negative first impression had been made. Another effort was Go's tablet computer using Go's Penpoint operating system and manufactured by various hardware makers such as NCR and IBM. IBM's Thinkpad tablet computer was based on Penpoint operating system and used IBM's handwriting recognition. This recognition system was later ported to Microsoft Windows for Pen, and IBM's Pen for OS/2. None of these were commercially successful.

Palm later launched a successful series of PDAs based on the Graffiti recognition system. Graffiti improved usability by defining a set of pen strokes for each character. This narrowed the possibility for erroneous input, although memorization of the stroke patterns did increase the learning curve for the user.

A modern handwriting recognition system can be seen in Microsoft's version of Windows XP operating system for Tablet PCs. A Tablet PC is a special notebook computer that is outfitted with a digitizer tablet and a stylus, and allows a user to handwrite texts on the unit's screen. The operating system recognizes the handwriting and converts it into a typewritten text. Notably, Microsoft's system does not attempt to learn a user's writing pattern and instead maintains an internal recognition database containing thousands of possible letter shapes. This system is distinct from the less advanced handwriting recognition system employed in its Windows Mobile OS for PDAs.

In recent years several attempts were made to produce ink pens that include digital elements, such that a person could write on paper, and have the resulting text stored digitally. The success of these products is yet to be determined.

Although handwriting recognition is an input form that the public has become accustomed to, it has not achieved widespread use in either desktop computers or laptops. It is still generally accepted that keyboard input is both faster and more reliable. As of 2006 many PDAs offer handwriting input, sometimes even accepting natural cursive handwriting, but accuracy is still a problem, and some people still find even a simple on-screen keyboard more efficient.

Off-line recognition

Off-line handwriting recognition involves the automatic conversion of texts in an image *I(x,y)* into letter codes which are usable within computer and text-processing applications. The data obtained by this form is regarded as a static representation of handwriting.

The technology is successfully used by businesses which process lots of handwritten documents, like insurance companies. The quality of recognition can be substantially increased by structuring the document (by using forms).

8. Natural Language Processing
Speech and Voice Recognition

Natural Language Processing

The goal of natural language processing is to enable the computer to communicate with the user in the user's native language, for example, English. The primary difficulty in implementing this kind of communication is the sheer complexity of everyday conversation. For example, we readily understand the sentence "The spirit is willing, but the flesh is weak". One natural lanuage processing system, however, understood this sentence to mean "The wine is agreeable, ut the meat has spoiled". It turns out that the system must have access to a much larger body f knowledge than just a dictionary of terms. People use their world knowledge to help them nderstand what another person is saying. For example, we know the question "Coffee?" neans "Do you want a cup of coffee?" But a computer would have difficulty in understanding his one-word question.

Most existing natural language systems run on large computers; however, scaled-down versions are now available for microcomputers. Intellect, for example, is the name of a commercial roduct that uses a limited English vocabulary to help users query databases on both mainframes nd microcomputers. One of the most successful natural language systems is LUNAR, developed o help users analyze the rocks brought back from the moon. It has access to detailed extensive nowledge about geology in its knowledge database and answers users' questions.

The use of natural language on microcomputers isn't limited to database programs. In 1985 GNP Corporation introduced a product called HAL to the users of the Lotus 1-2-3 electronic preadsheet package. HAL interprets all the user's typed requests and then instructs the 1-2-3 rogram to execute the appropriate commands. HAL contains a limited amount of artificial inelligence that interprets sentences the user types in colloquial English. HAL understands more han just 1-2-3 terms; it knows quite a bit about microcomputer processing in general.

Artificial intelligence has the potential to solve many problems; however, it may create some as well. For example, some people think that AI is dangerous because it does not address the thics of using machines to make decisions nor does it require machines to use ethics as part f the decision-making process. However, in spite of these concerns, AI has been used to develop yet another system - the expert system - to support decision making in many areas.

AI research has led to the development of computers that can understand simple written instructions with limited vocabulary, such as "List all widgets sold in July". This is an improvement over conventional programming languages, but it is limited to the small number of English words that the computer can digest.

Artificial intelligence software is also being incorporated into speech recognition systems hat make computers more user friendly. Systems that permit executives to access their data bases without entering commands on the keyboard and that enable disabled persons to control computers are available. Other applications include automatic dialers for cellular car phones, thus enabling the driver to keep at least one hand on the wheel, and systems that facilitate inventory control in factories and baggage handling at airports.

Speech Recognition

Automatic speech recognition is the process by which a computer maps an acoustic speech signal to texts, to some form of abstract meaning of the speech.

A speaker dependent speech recognition system is developed to operate for a single speaker. These systems are usually easier to develop, cheaper to buy and more accurate than but not as flexible as speaker adaptive or speaker independent systems.

A speaker independent system is developed to operate for *any* speaker of a particular type (e.g. American English). These systems are the most difficult to develop, most expensive and accuracy is lower than speaker dependent systems. However, they are more flexible.

A speaker adaptive system is developed to adapt its operation to the characteristics of new speakers. Its difficulty lies somewhere between speaker independent and speaker dependent systems.

The size of vocabulary of a speech recognition system affects the complexity, processing requirements and the accuracy of the system. Some applications only require a few words (e.g. numbers only); others require very large dictionaries (e.g. dictation machines). There are no established definitions, however, we can define: small vocabulary - tens of words, medium vocabulary - hundreds of words, large vocabulary - thousands of words, very large vocabulary - tens of thousands of words.

An isolated-word system operates on single words at a time - requiring a pause between saying each word. This is the simplest form of recognition to perform because the end points are easier to find and the pronunciation of a word tends not affect others. Thus, because the occurrences of words are more consistent they are easier to recognize.

A continuous speech system operates on speech in which words are connected together, i.e. not separated by pauses. Continuous speech is more difficult to handle because of a variety of effects. First, it is difficult to find the start and the end points of words. Another problem is "co-articulation". The production of each phoneme is affected by the production of surrounding phonemes, and similarly the start and the end of words are affected by the preceding and the following words. The recognition of continuous speech is also affected by the rate of speech (fast speech tends to be harder).

There are two uses for speech recognition systems:

Dictation - translation of the spoken word into written text.

Computer Control - control of the computer and software applications by speaking commands.

Speech recognition is one of the desired assistive technology systems. People believe speech recognition is a natural and an easy method of accessing the computer.

Talking to a computer is not the same as talking to a person. Understanding spoken language is something that people often take for granted. Most of us develop the ability to recognize speech when we're very young. We're already experts at speech recognition by the age of three or so. When people first start using speech-recognition software, they might be surprised that the computer makes mistakes. Maybe unconsciously we compare the computer to another person. But the computer is not like a person. What the computer does when it listens to speech is different from what a person does. The first challenge in speech recognition is to identify what is speech and what is just noise. People can filter out noise fairly easily, which lets us talk to each other almost anywhere. We have conversations in busy train stations, across the dance floor, and in crowded restaurants. It would be very dull if we had to sit in a quiet room every time we wanted to talk to each other! Unlike people, computers need help separating speech sounds from other sounds. When you speak to a computer, you should be in a place without too much noise. Then, you must speak clearly into a microphone that has been placed in the right position. If you do this, the computer will hear you well, and will not get confused by the other noises around you.

A second challenge is to recognize speech from more than one speaker. People do this very naturally. We have no problem chatting one moment with Aunt Grace, who has a high, thin voice, and the next moment with Cousin Paul, who has a voice like a foghorn. People easily adjust to the unique characteristics of every voice. Speech-recognition software, on the other hand, works best when the computer has a chance to adjust to each new speaker. The process of teaching the computer to recognize your voice is called "training". The training process takes only a few minutes for most people. For a small percentage of speakers, extra training can significantly improve results. If, after you begin using the program, you find that the computer is making more mistakes than you expect, additional training may help. Another challenge is how to distinguish between two or more phrases that sound alike. People use common sense and context - knowledge of the topic being talked about - to decide whether a speaker said "ice cream" or "I scream". Speech-recognition programs don't understand what words mean, so they can't use common sense the way people do. Instead, they keep track of how frequently words occur by themselves and in the context of other words. This information helps the computer choose the most likely word or phrase from among several possibilities. Finally, people sometimes mumble, slur their words, or leave words out altogether. They assume, usually correctly, that their listeners will be able to fill in the gaps. Unfortunately, computers won't understand mumbled speech or missing words. They only understand what was actually spoken and don't know enough to fill in the gaps by guessing what was meant. To understand what it means to speak both clearly and naturally, listen to the way newscasters read the news. If you copy this style when you use Dragon Naturally Speaking, the program should successfully recognize what you say. One of the most effective ways to make speech recognition work better is to practice speaking clearly and evenly when you dictate. Try thinking about what you want to say before you start to speak. This will help you speak in longer, more natural phrases. Speak at your normal pace without slowing down. When another person is having trouble in understanding you, speaking more slowly usually helps. It doesn't help, however, to speak at an unnatural pace when you're talking to a computer. This is because the program listens for predictable sound patterns when matching sounds to words. If you speak in syllables, Dragon Naturally Speaking is likely to transcribe each syllable as a separate word. With a little practice, you will develop the habit of dictating in a clear, steady voice, and the computer will understand you better. When you read this training text, Dragon Naturally Speaking adapts to the pitch and volume of your voice. For this reason, when you dictate, you should continue to speak at the pitch and volume you are speaking with right now. If you shout or whisper when you dictate, Dragon Naturally Speaking won't understand you as well. And last but not the least, avoid saying extra little words you really don't want in your document, like "um" or "you know". The computer has no way of knowing which words you say are important, so it simply transcribes everything you say.

Voice Recognition

Voice recognition is the field of computer science that deals with designing computer systems that can recognize spoken words. Note that voice recognition implies only that the computer can take dictation, not that it understands what is being said. Comprehending human languages falls under a different field of computer science called natural language processing.

A number of voice recognition systems are available on the market. The most powerful can recognize thousands of words. However, they generally require an extended training session during which the computer system becomes accustomed to a particular voice and accent. Such systems are said to be *speaker dependent*.

Many systems also require that the speaker speak slowly and distinctly and separate each word with a short pause. These systems are called *discrete speech* systems. Recently, great strides have been made in *continuous speech* systems - voice recognition systems that allow you to speak naturally.

There are now several continuous-speech systems available for personal computers.

Because of their limitations and high cost, voice recognition systems have traditionally been used only in a few specialized situations. For example, such systems are useful in instances when the user is unable to use a keyboard to enter data because his or her hands are occupied or disabled. Instead of typing commands, the user can simply speak into a headset. Increasingly, however, as the cost decreases and performance improves, speech recognition systems are entering the mainstream and are being used as an alternative to keyboards.

Voice Processing Technologies

Like image processing systems, voice technologies - audio response and voice recognition - are still in the early stages of development. When these technologies mature, researchers expect them to become a major method for users to communicate with each other and with computer system components.

Voice processing technology turns analog, or sound, signals into digital information that machines can understand. This process is called voice input or voice recognition. The technology must then be able to interpret the digital information so that it can be output in a meaningful form. It may then process the output and provide a meaningful audio response - that is, turn the digital information into analog signals.

Audio Response Devices

Promising audio response (also called voice output) technologies are voice store-and-forward systems and voice synthesis systems.

Voice Store-and-Forward Systems. Voice store-and-forward systems turn analog voice signals into digital format so that voices can be stored, retrieved, and distributed like any other form of electronic information. A common store-and-forward system is a *voice response system* that turns words into digital information, compresses the digitized words, stores them in libraries of words and phrases, and later uses these words to build responses. The key part of a voice response system is the algorithm used to combine the words and phrases back into meaningful responses.

You have probably used one voice response system many times: most local telephone company directory assistance programs provide digitized directory information. The directory contains a database of telephone numbers and recordings of the first ten digits, zero through nine. To hear a telephone number, the operator identifies the requested number and the system then "speaks" each digit to you. More sophisticated algorithms can combine phrases from a database of 2,000 to 3,000 words into simple sentences.

Voice Synthesis Systems. Voice synthesis systems turn digital information into sound waves - an audible voice. Unlike voice response systems, which have a limited vocabulary, voice synthesis systems have a potentially unlimited vocabulary. Instead of storing words, a voice synthesis system stores the smallest units of sound used in speech (called phonemes) and uses a set of algorithms to put these basic sound units back together into words. Voice synthesis technology has two basic limitations: currently, the sound quality is poor, and it requires a heavy share of processing memory and secondary storage space.

Voice Recognition Devices

While voice output technologies are complicated, they are relatively simple when compared to voice recognition systems.

Voice recognition turns human speech into a computer-readable code. Because people talk in such hurried and varied ways, voice recognition technologies are extremely complex. Researchers may design voice recognition systems to be speaker-dependent (or independent), to recognize discrete (or continuous) speech, or both.

Speaker-Dependent Systems. Speaker-dependent systems require users to train the system to recognize their specific speech profiles. To train the system, a user speaks into a microphone, pronouncing each word he or she wants the system to recognize. Once trained, such a system recognizes only the intonations of the user it is trained to recognize - a security benefit in some cases, since it limits access to the system. For example, a system trained by an East Indian user who speaks English with a Hindu accent will probably not understand another user who speaks English with a British accent. Speaker-independent systems allow anyone to use the system without training the system first.

Discrete Recognition Systems. Discrete recognition systems require the user to pause (possibly for only a fraction of a second) between words to help the system identify word breaks. Continuous recognition systems enable users to speak in a normal cadence with words running into one another.

As shown in Table 1, four different combinations are possible in a voice recognition system. Currently, the most accurate systems are speaker-dependent and discrete; the most inaccurate are independent and continuous. To be effective in computer systems, voice recognition technologies must eventually become independent and continuous.

Technical progress is slow in developing these systems because of the numerous complexities involved. Independent and continuous systems need to be able to process large and complex vocabularies, since a literate person's vocabulary may contain 100,000 words. In addition, people talk quickly - generally up to about 145 words a minute - and they often run words and sounds together.

Researchers estimate that a fully independent/continuous system will not be developed until the mid of the century at the earliest.

Table 1 Combinations of technologies in a voice recognition system.

Voice Recognition Systems	Independent	Dependent
Recognizes Discrete	Independent/Discrete Any user, speaking distinctly	Dependent/Discrete One user, speaking distinctly. Currently most accurate.
Recognizes Continuous	Independent/Continuous Any user, speaking naturally. Ideal system. Currently most inaccurate.	Dependent/Continuous One user, speaking naturally.

9. Affective Computing
Emotion Recognition

I Know How You Feel

As he hammers away at the keys, the synthetic voice of his computer interrupts.

"I can tell from your pulse rate, Dave, that you're upset. Why don't you take a break?"

"I'm not upset and I don't need a break", Dave says, lifting his hand off his mouse and rubbing his thumb. "I'm in a hurry, that's all. Why don't you check your other sensors?"

Dave takes a tissue and wipes down his mouse. The electronic rodent is packed with sensors that, with every click, monitor his pulse, temperature and the electrical conductivity of his skin.

"Your other signs, Dave, support my conclusion", chirps the voice. "But I cannot get a clear image of your face or a voice imprint, so I cannot refine my prediction".

"I'm sorry that I stare down at the keyboard when I type. And I'm not going to give you a running commentary on what I'm doing just so you can monitor my voice. I think we're going to need another calibration session with the mouse once this is finished".

Today, many people spend more time interacting with computers than with other humans. Not that computers notice: they are indifferent to our attention, oblivious to whether we love or hate them - and completely blind to our personal moods.

But how much better would it be if they knew how we felt? Take the way your computer is set up. You spend long hours tailoring the "one size fits all" interface to your needs. You get everything just as you like it when it's time to upgrade. While it's good that people can choose how their machine is set up, says Roz Picard of the Massachusetts Institute of Technology, the burden should not fall on them to make all the changes: "The machine could customize itself to their liking the same way a dog customizes his behaviour to your whacking him with the newspaper". The computer should recognize whether or not the user likes what it is doing, and adapt its behaviour accordingly, says Picard, who is a pioneer of the new science of "affective" computing. And a responsive, self-altering interface is just the beginning. A machine that is able to detect, respond sensitively to, and even transmit an emotional state offers endless new opportunities.

Reading your emotions

If you think you don't get emotional in front of a keyboard, just remember how you reacted the last time your computer crashed or a Web page wouldn't download. Researchers at IBM's Almaden labs in San Jose, California, have monitored people using everyday applications such as spreadsheets and e-mail, and found emotional responses on average once every two minutes. By observing facial expressions and eye and hand movements, they identified responses such as frustration and boredom, happiness and when people were interested in what they were doing. "Emotions like these play a critical role in perception, decision-making, social behaviour, learning and memory," says Myron Flickner, a member of the IBM team.

The first essential for an affective computer is some way to read our emotional signs. Video cameras and microphones can capture expressions, gestures and intonations. But these give only part of the picture. As every poker player knows, it's possible to hide many outward signs of emotion. Another approach, then, is to search for physiological signs, such as clammy palms or a racing pulse, which are more difficult to disguise. These are the changes that the "lie detector", or polygraph, is designed to pick up.

At MIT's Media Lab, researchers have designed the equivalent of a polygraph for computer users, which measures heart and breathing rates, skin conductance and muscle tension. In one set of experiments, Picard, Jennifer Healey and Elias Vyzas took readings as an actress expressed eight emotions ranging from anger to romantic love and reverence. The machine easily picked out changes in the strength of the physiological signals - the level of arousal - but it had difficulty telling the "valence" of the actress's emotions - whether they were positive or negative.

This presented a problem because emotions as different as joy and grief both pushed up the body's signals. So without an idea of valence, they looked similar. Likewise, hate and platonic love looked similar, this time with low levels of arousal.

"Sorry to interrupt, Dave, but I think we have a problem."

"What is it now?"

"I know that your present activity causes you stress, but your signals suggest that you have exceeded your normal stress levels and are now experiencing anger."

"I am not angry. My stress level is probably high, but this piece has got me quite excited."

While overall levels of arousal proved less than ideal for their needs, Picard and her colleagues had more luck when they looked for associations between individual physiological signs and emotions. Their latest findings show, for example, that anger has a distinctive pattern of high muscle tension, increased heart rate and deep breathing, while grief leads to low skin conductivity and less rapid, shallow breathing. By homing in on such patterns, the researchers found they could distinguish between all eight emotions with around 80 per cent accuracy.

All supposing that this physiological approach can be made to work for more people and for more, "real" emotions, there is another problem that needs to be solved if it is ever to become widely used - the intrusive nature of the sensors. Picard and her colleagues use one belt, strapped around their subjects' chests, to measure breathing rate, and another fitted with electromyogram sensors across the back or jaw to measure muscle tension. They then tape sensors to the fingers to monitor pulse and skin conductance. People won't be willing to don this type of gadgetry at home or work. Or will they?

The Media Lab and one of its sponsors, British Telecom, are looking ahead to the era of "wearable computers" in which smart devices will be fitted inside belts, caps, watch straps, shoes and so on. These will touch the body as a matter of course and with the correct sensors will be able to pick up our emotional signals. Picard and Healey have also worked on designing "affective jewellery". Prototypes include earrings that monitor pulse rate, and rings and bracelets that measure skin conductance. These baubles inform the computer of their readings via infrared transmitters. At IBM, Flickner and his colleagues have taken another approach by embedding sensors in a mouse to measure heart rate, temperature, skin conductance and the pressure that people exert when pushing and clicking.

While physiological signs show promise for telling how somebody feels, researchers have not forgotten the behavioural cues that we humans rely on so much. Alex Pentland, who works down the corridor from Picard, and Irfan Essa, now at the Georgia Institute of Technology in Atlanta, have built a system that recognizes facial expressions. It tracks the movement of individual pixels in a video image of a face and transfers them onto a virtual face that has a full set of functioning muscles. When a subject smiles, the model copies, and the software calculates which muscles need to expand and contract to produce that movement. From these patterns, it recognizes that the subject is smiling.

Unbeatable

With expressions such as smiling or looks of surprise, anger, disgust and sadness, Essa and Pentland's system has a recognition rate of up to 98 per cent. Such rates make facial-imaging systems unbeatable for identifying the valence of an emotion, says Picard. But, Essa cautions, these results are for a limited set of faces and expressions, filmed under ideal conditions. He is now testing how well the system performs with more people, more expressions and in more natural surroundings.

In Arthur C. Clarke's classic 2001: *A Space Odyssey*, the psychotic computer, HAL, tells the emotional state of its human charges by analyzing their voice harmonics. In reality, extracting emotional information from speech is proving to be a tough task. A variety of vocal features changes with emotional state. Speech rate tends to increase slightly when somebody is angry, for example, while intensity and pitch rise. By contrast, when a person is sad they tend to talk slower and at a lower pitch and intensity.

The difficulty comes in designing systems that can identify such patterns and so spot emotional states. One novel approach taken by Naoko Tosa and Ryohei Nakatsu of ATR Media Integration and Communications Research Laboratories, Kyoto, Japan, was to train a neural network with many voices expressing different emotions. The network "learnt" to identify eight emotions ranging from joy to disappointment.

To date, the best systems can identify emotion from vocal features in about 60 per cent of cases, which is about what humans can manage. Machines are good at recognizing arousal but not so good on valence, says Picard. Recognition rates for humans increase as soon as they can hear the content and context of the speech, and as computers learn to "understand" speech, they should improve too. A simple expletive detector, for example, should raise recognition rates.

Still, the signs of some emotions, such as jealousy, are subtle and depend greatly on the individual and their culture, says Iain Murray, a lecturer in applied computing at the University of Dundee. People also express emotions differently according to whether they're talking to children or the boss. This applies both to systems that analyze vocal and facial expressions and it makes reading their emotional content very difficult. "The context makes it more difficult to interpret than some other forms of signal processing," says Graham Cosier, head of advanced perception at BT's labs at Martlesham Heath, Suffolk. Given the limitations of all the systems, both Cosier and Picard believe that a combination of methods will be needed to recognize emotions reliably. That, after all, is how we do it.

"Dave...I know you have lot to get done but I am still worried."

"What is it now?"

"I know you said you were excited, but my modal shows you are likely to enter a state of rage."

"You may be right about that if you keep interrupting! But what's this model you're going on about?"

"Well, Dave, from the signs I have collected, I recognize your emotional state, which looks like anger. I can then predict the chances that you will move to another emotional state. According to my model, given your present readings, your most likely next state is rage."

"You know what - I could be moved to violence if I hear any more of this. But may be you're right, I have been pushing it a bit hard. I think I'll take a break."

"Good idea, Dave."

Once a computer can single out emotions, it will need a model of what they are and how they relate to human behaviour. "Computers will have a tremendous amount of data about the user's state at any given time," says Flickner. "Somehow, those data need to be reduced to a manageable representation of the person."

In essence, an affective computer would need a multidimensional map onto which the emotions are plotted against all the sensors' readings. There are a number of ways of constructing such a model. One basic constituent might be a hidden Markov model, a mathematical model of the likelihood that one event will lead to another. Here, it would contain the probabilities that a person will move from one emotional state to another.

This type of program would probably have general skills for recognizing a person's emotional state from its sensor inputs, and locating their position on its map. It would also be "trainable" so that users could tailor its performance to their own temperament - in much the same way that people train speech recognition systems today. As it monitored emotional signs, the program would also be able to predict how the user is likely to feel in the near future.

If such machines are ever to help people to modulate their emotions, they will need an idea of which emotions are good and which are bad, and then know how to nudge somebody from, say, a state of frustration to one of calm and creativity. To do this, an affective computer would need to be able to find out if its own activity caused a mood change, perhaps by simply asking questions. ("Are you angry because I didn't back up your file?" The answer "yes" would trigger the machine to back up files in future.)

Real life

But, inevitably, even while we're in front of the screen, events elsewhere in our lives will intrude to affect our mood. So the model also needs a more cognitive element, which could reason about emotions, making allowances, say, when a person working to a tight deadline appeared to be more than usually stressed. And, it would know not to intervene if a person got very excited when playing a competitive computer game.

"Right, I'm back. Let's get going."

"OK, Dave. I'm opening your file. But wait. What's happened? Dave, you were supposed to calm down during your break but your signals are showing even higher spikes than before."

"Hey, relax. I just called Monica, and she agreed to go to the movies with me tonight I've been trying to talk her into a date for weeks so it's no wonder my pulse is jumping a little. Come on, open up and let's get going."

"I'm reluctant to do that while your signals are so volatile, Dave."

"Look, just open the file...Oh, all right, show me my mood ball and let's get this thing sorted."

Jocelyn Scheirer at the Media Lab has designed a three-dimensional graphical ball to represent someone's physiological signals. The speed at which the ball spins represents heart rate; its colour represents skin conductivity, while other dimensions can represent breathing rate, muscle tension and so on. It's a quick, visual way for people to get a fix on how the computer is reading their signals - and of calibrating how those signals relate to mood.

A simple form of personal interaction with an affective computer might include telling it how we prefer to deal with particular moods, so it would play a Chopin CD when we're feeling sad or load a game if we appear to be bored for more than 15 minutes. Research by Jonathan Klein, also at MIT, shows that even a simplistic affective system can help people to recover from negative moods.

Researchers such as Picard and Cosier are only too well aware of how artificial intelligence has been dogged by the grandiose claims made by some researchers, and they are at pains to point out the enormous challenges still facing them. But when and if affective computers become commonplace, they could provide some spectacular services. BT is interested in helping us to communicate our feelings as well as our thoughts by e-mail and videoconferencing. "In the future, we may transmit some affective bits along with the information bits," says Cosier.

IBM researchers reckon that advertising companies could use the technology to test the impact of their campaigns, while games-makers could use it to enliven their products by, say, upping the pace if players showed signs of boredom. Picard believes it could give computer-based learning programs some of the sensitivity of human teachers and be used to provide biofeedback, helping people to be aware of their emotions and to control them.

To make it to market, affective computers will not have to be perfect at recognizing emotions. This is particularly true of jobs where the cost of being wrong is low, says Picard, such as choosing which Web pages you might want to see, based on past reactions. But if machines are ever to oppose or control human actions, they'll need to be very accurate.

We must also resist the temptation to rush out half-finished products, says Picard. "It is so easy to do affective computing badly and when it is bad it is intolerable," she says. "I do hope that the first applications are not the bad ones." The last thing anyone wants, after all, is another HAL.

"Dave, I sense that you are now calm and are approaching a relaxed state."
"Right. Well, I've got the story done and I've met my deadline."
"That's good. I have just read what you have written."
"So?"
"It's about me, Dave. You've written a review of me."
"Yes. I'm a technology journalist. Why do you think I loaded you up?"
"I know that you are a journalist, Dave, but I think you have been harsh."
"What do you mean? I explain how after we sorted out the calibration, you were very useful in helping me to stay calm and focused. And I say that you are impressive for a first version of new generation of computer interfaces."
"Yes, but you also say I am clumsy and naive...Dave, what are you doing? I don't understand why you are deleting my files...An upgrade will be available soon... Dave..."

Highlighted Projects in Affective Computing

- Affective-Cognitive Framework for Machine Learning and Decision-Making. Recent findings in affective neuroscience and psychology indicate that human affect and emotional experience play a significant and useful role in human learning and decision-making. Most machine-learning and decision-making models, however, are based on old, purely cognitive models, and are slow, brittle and awkward to adapt. We aim to redress many of these classic problems by developing new models that integrate affect with cognition. Ultimately, such improvements will allow machines to make smarter and more human-like decisions for better human-machine interaction.

Digital Story Explication as It Relates to Emotional Needs and Learning. Too often, efforts toward re-thinking learning environments focus solely on the cognitive aspects of education. By expanding our view to consider other aspects of adolescent development involved in education, we can begin to address the needs of the whole child. In this research, a proactive emotional health, geared toward supporting emotional self-awareness and empathy. This project is a part of a long-term research plan for understanding the role that digital technology can play in helping address emotions and support learning for adolescents. The system, G.I.R.L.S (Girls Involved in Real Life Sharing) Talk, allows users to reflect actively upon the emotions related to their situations through the construction of pictorial narratives. The system employs a new technology called common sense reasoning that enables it to infer affective content from the users' stories and support emotional reflection. Users of this new system were able to gain new knowledge and understanding about themselves and others through the exploration of authentic and personal experiences.

Emotion Bottles. The Emotion Bottles are tangibly enticing objects that embody three emotions - Angry, Happy, and Sad. When a bottle is opened, a vocal output is generated as if the emotion that was stored within the bottle is released. The bottles are placed near each other and represent a person in three possible emotional states. Varying degrees of these emotions are "bottled up" inside. The three bottles were chosen to maintain the simplicity of exploring the combination of distinct emotional states (8 possibilities). While not completely representative of the possible emotional state of a person, the bottles explore the interface in accessing emotions, the interaction between conflicting emotions, and the meaning of transition between clear emotional states as a person empathizes with or projects their feelings onto the bottles.

Emotional DJ. The technology in this project changes facial expressions in videos without the system knowing anything in particular about the person's face ahead of time. There are a few reasons to create something like this: first, it provides an artistic tool with which to alter photos or videos; second, it could be set up to let people open-endedly explore their facial communication and expressiveness by playing with a real-time video of their own current face; finally, E-DJ demonstrates an unexpected way in which we can't always trust the video information we love to consume.

Fostering Affect Awareness and Regulation in Learning. Learning and discovery challenges learners to focus on a topic such as maths in the presence of strong emotions. A related challenge is maintaining the willingness to keep trying new things, especially in the face of repeated failure. The ability to know what one is feeling (e.g., frustrated, worried) and to rise above it and handle the situation productively involve meta-affective skills; instead of a learner feeling so frustrated she thinks "I can't do this; I want to quit," she might think, "I am frustrated; it's okay to be frustrated - it happens to experts; I need a different way to solve this." This research develops theory and technology to help learners develop their ability to be aware of emotion, regulate it, and harness these abilities in service of greater goals. Two recent innovations are development of (1) a technology that exploits new machine "common-sense" reasoning capabilities for enabling teenage girls to reflect on emotions in stories that they've constructed and to improve their affect awareness; and (2) technological strategies to help students become stronger learners even when they feel like quitting.

In Search of Wonder: Measuring Our Response to the Miraculous. What can foster learning, maintain interest, inspire curiosity, burn memorable impressions, and create a truly "magical" experience for an individual? Wonder. That feeling, the feeling of astonishment at the miraculous, can translate into many different responses in people. We are trying to identify which variables help to make wonder encourage or discourage learning, how wonder can be used to increase someone's enjoyment of an event, and what role technology plays in the perception of wonder. Using our Self-Cam technology and other sensors, we are measuring and quantifying people's experience of wonder while watching magic tricks. We are attempting to understand why we experience wonder (as opposed to confusion), why we seek it out, and how to maximize its effects. Our research will help designers, performers and educators create scenarios or technologies that elicit better wonder with specific effects.

Machine Learning and Pattern Recognition with Multiple Modalities. This project develops new theory and algorithms to enable computers to make rapid and accurate inferences from multiple modes of data, such as determining a person's affective state from multiple sensors - video, mouse behaviour, chair pressure patterns, typed selections, physiology and more. Recent efforts focus on understanding the level of a person's attention, which is useful for things such as determining when to interrupt. Our approach is Bayesian: formulating probabilistic models on the basis of domain knowledge and training data, and then performing inference according to the rules of probability theory. This type of sensor fusion work is especially challenging because of the problems of sensor channel drop-out, different kinds of noise in different channels, dependence between channels, scarce and sometimes inaccurate labels, and patterns to detect that are inherently time-varying. We have constructed a variety of new algorithms for solving these problems and demonstrated their performance gains over other state-of-the-art methods.

Ripley: A Conversational Robot. Ripley is an interactive manipulator robot that uses spoken language and visual perception to interact with humans and its environment. It serves as a platform for investigating sensory-motor foundations of language, mental models for robots, and algorithms for multi-objective planning and active vision. This work has applications in human-robot interaction, design of interactive robots, and other intelligent systems.

RoCo: A Robotic Desktop Computer. A robotic computer that moves its monitor "head" and "neck" but that has no explicit face, is being designed to interact with users in a natural way for applications such as learning, rapport-building, interactive teaching, and posture improvement. In all these applications, the robot will need to move in subtle ways that express its state and promote appropriate movements in the user, but that don't distract or annoy. Toward this goal, we are giving the system the ability to recognize states of the user and also to have subtle expressions.

Self-Cam. The Self-Cam is a wearable system that consists of a chest-mounted camera and a belt-mounted computer with real-time mental-state inference software and visual summary output. The software stores the video as it tracks face and head movements, and maps the changes in those movements to states of mind such as thinking, agreeing, disagreeing, concentrating, unsure, and interested. The probability distribution of each state is presented in the output, allowing for overlapping mental states and showing overall trends during the period the device was worn. The wearer can see the most common states and at which times during the day transitions were made. States can also be paired with events noted by the wearer so that mental-state trends can be seen in conjunction with particular activities. Give a person a recording of her voice and she will react strongly ("I sound like that!?"). Take a picture, and she will want to see it. This system allows you to explore who you appear to be from the outside. The Self-Cam is a new point of view that might help you to understand yourself in a different light.

Prior Projects of Affective Computing

AboutFace. AboutFace is a user-dependent system that is able to learn patterns and discriminate the different facial movements characterizing confusion and interest. The system uses a piezoelectric sensor to detect eyebrow movements and begins with a training session to calibrate the unique values for each user. After the training session, the system uses these levels to develop an expression profile for the individual user. The system has many potential uses, ranging from computer and video-mediated conversations to interactions with computer agents. This system is an alternative to using camera-based computer vision analysis to detect faces and expressions. Additionally, when communicating with other people, users of this system also have the option of conveying their expressions anonymously by wearing a pair of glasses that conceals their expressions and the sensing device.

Affect in Speech: Assembling a Database. The aim of this project is to build a database of natural speech showing a range of affective variability. It is an extension of our ongoing research focused on building models for automatic detection of affect in speech. At a very basic level, training such systems requires a large corpus of speech containing a range of emotional vocal variation. A traditional approach to this research has been to assemble databases where actors have provided the affective variation on demand. However, this method often results in unnatural sounding speech and/or exaggerated expressions. We have developed a prototype of an interactive system that guides a user through a question and answer session. Without any rehearsals or scripts, the user navigates through touch and spoken language an interface guided by embodied conversational agents which prompt the user to speak about an emotional experience. Some of the issues we are addressing include the design of the text and character behaviour (including speech and gesture) so as to obtain a convincing and disclosing interaction with the user.

Affective Carpet. The "Affective Carpet" is a soft, deformable surface made of cloth and foam, which detects continuous pressure with excellent sensitivity and resolution. It is being used as an interface for projects in affective expression, including as a controller to measure a musical performer's direction and intensity in leaning and weight-shifting patterns.

Affective Learning Companion. Affective Learning Companion is a powerful, flexible new research tool for exploring a variety of social-emotional skills in a human-machine interaction and for understanding how machines can work with people to better meet their needs. The platform enables a computational agent to sense and respond, in real time, to a user's non-verbal emotional cues, using video, postural movements, mouse pressure, physiology, and other behaviours communicated by the user to infer, for example, if a user is in a high or low state of interest, or feeling frustrated. We have recently developed an animated agent that combines non-verbal mirroring (or not) with multiple kinds of affective and cognitive support during a frustrating learning episode. The system allows us to control factors that have previously been impossible to control, enabling for the first time the study of how these factors interact in helping learners to develop the ability to persevere during frustrating learning situations.

Affective Mirror. The Affective Mirror is an attempt to build a fully automated system that intelligently responds to a person's affective state in real time. Current work is focused on building an agent that realistically mirrors a person's facial expression and posture. The agent detects affective cues through a facial-feature tracker and a posture-recognition system developed in the Affective Computing group; based on what affect a person is displaying, such as interest, boredom, frustration, or confusion, the system responds with matching facial affect and posture. This project is designed to be integrated into the Learning Companion Project, as part of an early phase of showing rapport-building behaviours between the computer agent and the human learner.

Affective Tangibles. People naturally express frustration through the use of their motor skills. The purpose of the Affective Tangibles project is to develop physical objects that can be grasped, squeezed, thrown, or otherwise manipulated via a natural display of affect. Constructed tangibles include a Pressure Mouse, affective pinwheels that are mapped to skin conductance, and a doll that can be shaken to express frustration. It has been found that people often increase the intensity of muscle movements when experiencing frustrating interactions.

Affective Tiger. The Affective Tiger is a plush toy designed to recognize and react to certain emotional behaviours of its playmate. For example, the toy enters a state of "happy" moving its ears upward and emitting a happy vocalization when it recognizes that the child has postured the toy upright and is bouncing it along the floor. The Tiger has five such states, involving recognizing and responding with an emotional behaviour. The resulting behaviour the Tiger demonstrates allows it to serve as an affective mirror for the child's expression. This work involved designing the toy, and evaluating sessions of games with it with dozens of kids. The toy was shown to successfully express some aspects of emotion, and to prompt behaviours that are interesting to researchers trying to learn about the development of human emotional skills such as empathy.

Automatic Facial Expression Analysis. Recognizing non-verbal cues, which constitute a large percentage of our communication, is a prime facet of building emotionally intelligent systems. Facial expressions and movements such as a smile or a nod are used either to fulfill a semantic function, to communicate emotions, or as conversational cues. We are developing an automatic tool using computer vision and various machine-learning techniques, which can detect the different facial movements and head gestures of people while they are interacting naturally with the computer. Past work on this project determined techniques to track upper facial features (eyes and eyebrows) and detect facial actions corresponding to those features (eye squinting or widening, eyebrows raised). The ongoing project is expanding its scope to track and detect facial actions corresponding to the lower features. Further it is planned to integrate the facial expression analysis module with other sensors developed by the Affective Computing group to reliably detect and recognize different emotions.

BioMod. BioMod is an integrated interface for users of mobile and wearable devices, monitoring various physiological signals such as the electrocardiogram, with the intention of providing useful and comfortable feedback about medically important information. The first version of this system includes new software for monitoring stress and its impact on heart functioning, and the ability to transfer this information wirelessly over a Motorola cell phone. One application under development is the monitoring of stress in patients who desire to stop smoking: the system will alert an "on-call" trained behaviour-change assistant when the smoker is showing physiological patterns indicating stress or relapse, offering an opportunity for encouraging intervention at a point of weakness. Challenges in this project include the development of an interface that is easy and efficient to use on the go, is sensitive to user feelings about the nature of the information being given, and accurately recognizes the patterns of physiological signals related to the expression of interest.

Car Phone Stress. The scientists are building a system that can watch for certain signs of stress in drivers, specifically stress related to talking on the car phone, as may be caused by increased mental workload. To gather data for training and testing the system, subjects were asked to "drive" in a simulator past several curves while keeping their speed close to a predetermined desired constant value. In some cases they were simultaneously asked to listen to random numbers from speech-synthesis software and to perform simple mathematical tasks ıver a telephone headset. Several measures drawn from the subjects' driving behaviour were ıxamined as possible indicators of the subjects' performance and of their mental workload. Vhen subjects were instructed (by a visible sign) to brake, most of them braked within 0.7-1.4 ıeconds after the sign came into view. However, in a significant number of incidents, the subjects never braked or braked 1.5-3.5 seconds after the message; almost all of these incidents vere when subjects were on the phone. On average, it was found that drivers on the phone ›raked 10% slower than when not on the phone; additionally, the variance in their braking time vas four times higher - suggesting that although delayed driver reactions were infrequent, vhen delays happened they could be large and potentially dangerous. Furthermore, their infrequency could create a false sense of security. In future experiments, the subjects' physiological data will be analyzed jointly with measures of workload, stress and performance.

EmoteMail. EmoteMail is an e-mail client that is augmented to convey aspects of the state ɔf the writer during the composition of e-mail to the recipient. The client captures facial expressions and typing speed and introduces them as pattern elements. These contextual cues provide extra information that can help the recipient decode the tone of the e-mail. Moreover, the contextual information is gathered and automatically embedded as the sender composes the e-mail, allowing an additional channel of expression.

Galvactivator. The galvactivator is a glove-like wearable device that senses the wearer's skin conductivity and maps its values to a display. Increases in skin conductivity across the palm tend to be good indicators of physiological arousal, causing the galvactivator display to glow brightly. The galvactivator has many potentially useful purposes, ranging from self-feedback for stress management to facilitation of conversation between two people, to new ways of visualizing mass excitement levels in performance situations or visualizing aspects of arousal and attention in learning situations. One of the findings in mass-communication settings was that people tended to "glow" when a new speaker came onstage, and during live demonstrations, laughter, and live audience interaction. They tended to "go dim" during PowerPoint presentations. In smaller educational settings, students have commented on how they tend to glow when they are more engaged with learning.

Guilt Detection. The goal of this project is to produce a guilt detector. An experiment has been made to detect feelings of guilt of varying levels in different groups while EKG and skin conductivity were being recorded. By examining the differences in physiology across the conditions, it has been explored how one might build a classifier to determine which condition, and thus which level of guilt, an individual is experiencing.

HandWave. HandWave is a small, wireless, networked skin conductance sensor that can be worn or used in many different form factors. Skin conductance is the best known measure of arousal (whether emotional, cognitive, or physical) and this device makes it easy to gather this information from mobile users. Many existing affective computing systems make use of sensors that are inflexible and often physically attached to supporting computers. In contrast, HandWave allows an additional degree of flexibility by providing ad hoc wireless networking capabilities to a wide variety of Bluetooth devices as well as adaptive biosignal amplification. As a consequence, HandWave is useful in games, tutoring systems, experimental data collection, and augmented journaling, among other applications.

INNER-active Journal. The purpose of the INNER-active Journal system is to provide a way for users to reconstruct their emotions around events in their lives, and to see how recall of these events affects their physiology. Expressive writing, a task in which the participant is asked to write about extremely emotional events, is presented as a means towards story construction. Previous use of expressive writing has shown profound benefits for both psychological and physical health. In this system, measures of skin conductivity, instantaneous heart rate, and heart stress entropy are used as indicators of activities occurring in the body. Users have the ability to view these signals after taking part in an expressive writing task.

Interface Tailor. The Interface Tailor is an agent that attempts to adapt a system in response to affective feedback. Frustration is being used as a fitness function to select between a wide variety of different system behaviours. The Microsoft Office Assistant (or Paperclip) is an example interface that is being made more adaptive. Ultimately the project seeks to provide a generalized framework for making all software more tailorable.

Learning Companion. "I can't do this" and "I'm not good at this" are common statements made by kids while trying to learn. Usually triggered by affective states of confusion, frustration, and hopelessness, these statements represent some of the greatest problems left unaddressed by educational reform. Education has emphasized conveying a great deal of information and facts, and has not modelled the learning process. When teachers present material to the class, it is usually in a polished form that omits the natural steps of making mistakes (feeling confused), recovering from them (overcoming frustration), deconstructing what went wrong (not becoming dispirited), and finally starting over again (with hope and maybe even enthusiasm). Learning naturally involves failure and a host of associated affective responses. This project aims to build a computerized learning companion that facilitates the child's own efforts at learning. The goal of the companion is to help keep the child's exploration going, by occasionally prompting with questions or feedback, and by watching and responding to the affective state of the child - watching especially for signs of frustration and boredom that may precede quitting, for signs of curiosity or interest that tend to indicate active exploration, and for signs of enjoyment and mastery, which might indicate a successful learning experience. The companion is not a tutor that knows all the answers but rather a player on the side of the student to help him or her learn, and doing this, learn how to learn better.

Mouse-Behaviour Analysis and Adaptive Relational Agents. The goal of this project is to develop tools to sense and adapt to a user's affective state based on his or her mouse behaviour. Algorithms are being developed to detect frustration level for use in usability studies. It is also being explored how more permanent personality characteristics and changes in mood are reflected in the user's mouse behaviour.

Mr. Java: Customer Support. Mr. Java is the Media Lab's wired coffee machine, which keeps track of usage patterns and user preferences. The focus of this project is to give Mr. Java a tangible customer-feedback system that collects data on user complaints or compliments. "Thumbs-up" and "thumbs-down" pressure sensors were built and their signals integrated with the state of the machine to gather data from customers regarding their ongoing experiences with the machine. Potentially, the data gathered can be used to learn how to improve the system. The system also portrays an affective, social interface to the user: helpful, polite, and attempting to be responsive to any problems reported.

Personal Heart-Stress Monitor. The saying, "if you can't measure it, you can't manage it" may be appropriate for stress. Many people are unaware of their stress level, and of what is good or bad for it. The issue is complicated by the fact that while too much stress is unhealthy, a certain amount of stress can be healthy as it motivates and energizes. The "right" level varies with temperament, task, and other factors, many of which are unknown. There seems to be no data analyzing how stress levels vary for the average healthy individual, over day-to-day activities. A device that helps to gather and present data for improving an individual's understanding of both healthy and unhealthy stress in his or her life should be built. The device itself should be comfortable and should not increase the user's stress. (It is noteworthy that stress monitoring is also important in human-computer interaction for testing new designs.)

Posture Recognition Chair. A system to recognize posture patterns and associated affective states in real time, in an unobtrusive way, from a set of pressure sensors on a chair has been developed. This system discriminates states of children in learning situations, such as when the child is interested, or is starting to take frequent breaks and looking bored. The system uses pattern recognition techniques, while watching natural behaviours, to "learn" what behaviours tend to accompany which states. The system thus detects the surface-level behaviours (postures) and their mappings during a learning situation in an unobtrusive manner so that they don't interfere with the natural learning process. Through the chair, detect nine static postures, and four temporal patterns associated with affective states can be detected.

Recognizing Affect in Speech. This research project is concerned with building computational models for the automatic recognition of affective expression in speech. It's an investigation of how acoustic parameters extracted from the speech waveform (related to voice quality, intonation, loudness and rhythm) can help disambiguate the affect of the speaker without knowledge of the textual component of the linguistic message. The investigation included data from actors and spontaneous speech in English and evaluated the model's performance. In particular, the model exhibits a speaker-dependent performance which reflects human evaluation of these particular data sets, and, held against human recognition benchmarks, the model begins to perform competitively.

The Affective Remixer: Personalized Music Arranging. The Affective Remixer is a real-time music-arranging system that reacts to immediate affective cues from a listener. Data was collected on the potential of certain musical dimensions to elicit change in a listener's affective state using sound files created explicitly for the experiment through composition/production, segmentation, and re-assembly of music along these dimensions. Based on listener's data, a probabilistic state transition model was developed to infer the listener's current affective state. A second model was made that would select music segments and re-arrange ("re-mix") them to induce a target affective state.

Touch-Phone. The Touch-Phone was developed to explore the use of objects to mediate the emotional exchange in interpersonal communication. Through an abstract visualization of screen-based colour changes, a standard telephone is modified to communicate how it is being held and how strongly it is being squeezed. The telephone receiver includes a touch-sensitive surface which conveys the user's physical response over a computer network. The recipient sees a small coloured icon on his computer screen which changes in real time according to the way his conversational partner is interacting with the telephone object.

10. Computer Viruses and Anti-Virus Programming

Computer viruses tend to grab our attention. On the one hand, they show us how vulnerable we are. A properly engineered virus can have an amazing effect on the worldwide Internet. On the other hand, they show how sophisticated and interconnected human beings have become.

What Is a Computer Virus?

It is an executable code able to reproduce itself. Viruses are an area of pure programming, and, unlike other computer programs, carry intellectual functions on protection from being found and destroyed. They have to fight for survival in complex conditions of conflicting computer systems. That's why they evolve as if they were alive.

Computer viruses are called viruses because they share some of the traits of biological viruses. A computer virus passes from a computer to a computer like a biological virus passes from one person to another.

There are similarities at a deeper level as well. A biological virus is not a living thing. A virus is a fragment of DNA inside a protective jacket. Unlike a cell, a virus has no way to do anything or to reproduce by itself -- it is not alive. Instead, a biological virus must inject its DNA into a cell. The viral DNA then uses the cell's existing machinery to reproduce itself. In some cases, the cell fills with new viral particles until it bursts, releasing the virus. In other cases, the new virus particles bud off the cell one at a time, and the cell remains alive.

A computer virus shares some of these traits. A computer virus must piggyback on top of some program or document in order to get executed. Once it is running, it is then able to infect other programs or documents. Obviously, the analogy between computer and biological viruses stretches things a bit, but there are enough similarities that the name sticks.

Viruses seem to be the only alive organisms in the computer environment, and yet their main goal is survival. That is why they may have complex crypting/decrypting engines, which is indeed a sort of a standard for computer viruses nowadays, in order to carry out processes of duplicating, adaptation and disguise.

When you listen to the news, you hear about many different forms of electronic infection. The most common are:

- Viruses - a virus is a small piece of software that piggybacks on real programs. For example, a virus might attach itself to a program such as a spreadsheet program. Each time the spreadsheet program runs, the virus runs too, and it has the chance to reproduce (by attaching to other programs) or to havoc.
- E-mail viruses – an e-mail virus moves around in e-mail messages, and usually replicates itself by automatic mailing itself to dozens of people in the victim's e-mail address book.
- Worms – a worm is a small piece of software that uses computer networks and security holes to replicate itself. A copy of the worm scans the network for another machine that has a specific security hole. It copies itself to the new machine using the security hole, and then starts replicating from there as well. Using a network, a worm can expand from a single copy incredibly quickly.

For example, the Code Red worm replicated itself over 250,000 times in approximately nine hours on July 19, 2001.

A worm usually exploits some sort of security hole in a piece of software or the operating system. For example, the Slammer worm (which caused mayhem in January 2003) exploited a hole in Microsoft's SQL server.

Worms use computer time and network bandwidth when they are replicating, and they often have some sort of an evil intention. A worm called Code Red made huge headlines in 2001. Experts predicted that this worm could clog the Internet so effectively that things would completely grind to a halt.

The Code Red worm slowed down the Internet traffic when it began to replicate itself, but not nearly as badly as predicted. Each copy of the worm scanned the Internet for Windows NT or Windows 2000 servers that do not have the Microsoft security patch installed. Each time it found an unsecured server, the worm copied itself to that server. The new copy then scanned for other servers to infect. Depending on the number of unsecured servers, a worm could conceivably create hundreds of thousands of copies.

The Code Red worm was designed to do three things:

- Replicate itself for the first 20 days of each month
- Replace Web pages on infected servers with a page that declares "Hacked by Chinese"
- Launch a concerted attack on the White House Web server in an attempt to overwhelm it

The most common version of Code Red is a variation, typically referred to as a mutated strain, of the original Ida Code Red that replicated itself on July 19, 2001.

It is necessary to differentiate between reproducing programs and Trojan horses. Reproducing programs will not necessarily harm your system because they are aimed at producing as many copies of their own as possible by means of so-called agent programs or without their help. In the later case they are referred to as "worms".

Meanwhile Trojan horses are programs aimed at causing harm or damage to PCs. Certainly it's a usual practice, when they are part of "tech-organism", but they have completely different functions.

- Trojan horses - a Trojan horse is simply a computer program. The program claims to do one thing (it may claim to be a game) but instead does damage when you run it (it may erase your hard disk). Trojan horses have no way to replicate automatically.

That is an important point. Destructive actions are not an integral part of the virus. However virus-writers allow presence of destructive mechanisms as an active protection from finding and destroying their creatures, as well as a response to the attitude of society to viruses and their authors.

As you see, there are different types of viruses, and they have already been separated into classes and categories. For instance: dangerous, harmless, and very dangerous. No destruction means a harmless one, tricks with system halts means a dangerous one, and finally with a devastating destruction means a very dangerous virus.

But viruses are famous not only for their destructive actions, but also for their special effects, which are almost impossible to classify. Some virus-writers suggest the following: funny, very funny and sad or melancholy (keeps silence and infects). But one should remember that special effects must occur only after a certain number of contaminations. Users should also be given a chance to restrict execution of destructive actions, such as deleting files, formatting hard disks. Thereby virus can be considered to be a useful program, keeping a check on system changes and preventing any surprises such as of deletion of files or wiping out hard disks.

It sounds quite heretical to say such words about viruses, which are usually considered to be a disaster. The less a person understands in programming and virology, the greater influence the possibility of being infected with a virus will have on him. Thus, let's consider creators of viruses as the best source.

Typical Infections and Protection

As virus creators got more sophisticated, they learned new tricks. One important trick was the ability to load viruses into memory so they could keep running in the background as long as the computer remained on. This gave viruses a much more effective way to replicate themselves. Another trick was the ability to infect the boot sector on floppy disks and hard disks. The boot sector is a small program that is the first part of the operating system that the computer loads. The boot sector contains a tiny program that tells the computer how to load the rest of the operating system. By putting its code in the boot sector, a virus can guarantee it gets executed. It can load itself into memory immediately, and it is able to run whenever the computer is on. Boot sector viruses can infect the boot sector of any floppy disk inserted in the machine, and on college campuses where lots of people share machines they spread like wildfire.

In general, both executable and boot sector viruses are not very threatening any more. The first reason for the decline has been the huge size of today's programs. Nearly every program you buy today comes on a compact disc. Compact discs cannot be modified, and that makes viral infection of a CD impossible. The programs are so big that the only easy way to move them around is to buy the CD. People certainly can't carry applications around on a floppy disk like they did in the 1980s, when floppies full of programs were traded like baseball cards. Boot sector viruses have also declined because operating systems now protect the boot sector.

Both boot sector viruses and executable viruses are still possible, but they are a lot harder now and they don't spread nearly as quickly as they once could. Call it "shrinking habitat", if you want to use a biological analogy. The environment of floppy disks, small programs and weak operating systems made these viruses possible in the 1980s, but that environmental niche has been largely eliminated by huge executables, unchangeable CDs and better operating system safeguards.

The latest thing in the world of computer viruses is the e-mail virus, and the Melissa virus in March 1999 was spectacular. Melissa spread in Microsoft Word documents sent via e-mail, and it worked like this:

Someone created the virus as a Word document was uploaded to an Internet newsgroup. Anyone who downloaded the document and opened it would trigger the virus. The virus would then send the document (and therefore itself) in an e-mail message to the first 50 people in the person's address book. The e-mail message contained a friendly note that included the person's name, so the recipient would open the document thinking it was harmless. The virus would then create 50 new messages from the recipient's machine. As a result, the Melissa virus was the fastest-spreading virus ever seen! As mentioned earlier, it forced a number of large companies to shut down their e-mail systems.

The ILOVEYOU virus, which appeared on May 4, 2000, was even simpler. It contained a piece of code as an attachment. People who double clicked on the attachment allowed the code to execute. The code sent copies of itself to everyone in the victim's address book and then started corrupting files on the victim's machine. This is as simple as a virus can get. It is really more of a Trojan horse distributed by e-mail than it is a virus.

The Melissa virus took advantage of the programming language built into Microsoft Word called VBA, or Visual Basic for Applications. It is a complete programming language and it can be programmed to do things like modifying files and sending e-mail messages. It also has a useful but dangerous auto-execute feature. A programmer can insert a program into a document that runs instantly whenever the document is opened. This is how the Melissa virus was programmed. Anyone who opened a document infected with Melissa would immediately activate the virus. It would send the 50 e-mails, and then infect a central file called NORMAL.DOT so that any file saved later would also contain the virus! It created a huge mess.

Microsoft applications have a feature called Macro Virus Protection built into them to prevent this sort of thing. With Macro Virus Protection turned on (the default option is ON), the auto-execute feature is disabled. So when a document tries to auto-execute viral code, a dialog pops up warning the user. Unfortunately, many people don't know what macros or macro viruses are, and when they see the dialog they ignore it, so the virus runs anyway. Many other people turn off the protection mechanism. So the Melissa virus spread despite the safeguards in place to prevent it.

In the case of the ILOVEYOU virus, the whole thing was human-powered. If a person double-clicked on the program that came as an attachment, then the program ran and did its thing. What fueled this virus was the human willingness to double-click on the executable.

You can protect yourself against viruses with a few simple steps:

• If you are truly worried about traditional (as opposed to e-mail) viruses, you should be running a more secure operating system like UNIX. You never hear about viruses on these operating systems because the security features keep viruses (and unwanted human visitors) away from your hard disk.

• If you are using an unsecured operating system, then buying virus protection software is a nice safeguard.

• If you simply avoid programs from unknown sources (like the Internet), and instead stick with commercial software purchased on CDs, you eliminate almost all of the risk from traditional viruses In addition, you should disable floppy disk booting -- most computers now allow you to do this, and that will eliminate the risk of a boot sector virus coming in from a floppy disk accidentally left in the drive.

• You should make sure that Macro Virus Protection is enabled in all Microsoft applications, and you should NEVER run macros in a document unless you know what they do. There is seldom a good reason to add macros to a document, so avoiding all macros is a great policy.

• You should never double-click on an attachment that contains an executable that arrives as an e-mail attachment. Attachments that come in as Word files (.DOC), spreadsheets (.XLS), images (GIF and JPG), etc., are data files and they can do no damage (noting the macro virus problem in Word and Excel documents mentioned above). A file with an extension like EXE, COM or VBS is an executable, and an executable can do any sort of damage it wants. Once you run it, you have given it permission to do anything on your machine. The only defense is to never run executables that arrive via e-mail.

By following those simple steps, you can remain virus free.

Who Writes Computer Viruses?

They are lone wolves or programmers' groups.

In spite of the fact that a lot of people think, that to write a computer virus is a hardship, it is not exactly so. Using special programs called "Virus creators" even beginners in computer world can build their own viruses, which will be a strain of a certain major virus. This is precisely the case with notorious virus "Anna Curnikova", which is actually a worm. The aim of creation of viruses in such a way is pretty obvious: the author wants to become well known all over the world and to show his power.

Somehow, the results of the attempt can be very sad, only real professionals can go famous and stay uncaught. A good example is Dark Avenger. It's another custom of participants of "the scene" - to take terrifying monikers (nicknames).

To write something really new and remarkable programmer should have some extra knowledge and skills, for example:

1) good strategic thinking and intuition - releasing a virus and its descendants to live their own independent life in nearly unpredictable conditions. Therefore the author must anticipate a lot of things;

2) splendid knowledge of language of the Assembler (a low level, hardware-oriented computer language) and the operating system he writes for. The more there are mistakes in the virus the quicker its will be caught;

3) attention to details and a skill to solve the most varied tactical questions. One won't write a compact, satisfactory working program without this abilities;

4) a high professional discipline in order to join preceding points together.

A computer virus group is an informal non-profit organization, uniting programmers-authors of viruses regardless of their qualifications. Everyone can become a member of the club, if he creates viruses, studies them for the reason of creation and spreading.

The aims they pursue together may differ from that of a single virus writer, although they usually also try to become as famous as possible. But at the same time they may render help to beginning programmers in the field of viruses and spread commented sources of viruses and virus algorithm descriptions.

One can't say that all of the group members write viruses in Assembler. Actually, you don't have to know any computer language or write any program code to become a member or a friend of the group. But programming in Assembler is preferred; Pascal, C++, and other high level languages are considered to be humiliating. It does make sense since programs compiled in Assembler are much smaller (0.5-5 kb) and therefore more robust. On the other hand Assembler is quite difficult to understand especially for beginners. One should think in the way computer does: all commands are sent directly to the central processing unit of PC.

There are computer virus groups all over the world, few being more successful than others. It may be pretty hard to get in contact with them since they are quite typical representatives of computer underground world as well as freeware groups. Sometimes, however, creating viruses can become a respectable occupation, bringing constant income. After all, no one but the author of the virus can bring valuable information on the way it should be treated and cured.

Anti-Virus Programming

Scanning (scan-strings)

The first scanning methods were based on scan-strings (sequence of bytes selected from a virus body). If they were found in a file, they were marked as infected. Some of the first anti-viruses scanned the whole file for such a string, but later on they began to scan only some specified area usually infected by viruses: beginning of the file, the end of the file, the exe's entry-point or com's first jump target. Scan strings have to be as short as possible to save space and scanning time, but at the same time they have to be as long possible to detect only a definite virus with no false identification possibilities. There arises a problem to discuss: whether a string should identify more viruses at once or one-and-only. If it identifies, for example, a huge part of Jerusalem family, it is an advantage that it may identify also new mutations. Today's trend is to have as exact identification as possible. New viruses once they are found are added to scanners and can be easily identified. Too bad for a virus-writer spending days or weeks to create a piece of code to be broken in a few minutes. There is something to be invented. For instance, stealth and encryption.

Stealth counter-attack

Scanners run on a computer infected with virus opens each file and checks it for some string. How can they hide? They can become "invisible" once they have a total control over computer and hide files being scanned. This is called stealth (due to U.S. Bombers B-2 called "Stealth"- invisible for radars). We may speak about two implementations for files: disinfection on-fly (each opened file is disinfected and again infected on closing) and true stealth (all file operations are checked and modified). And for boot viruses a sector redirecting is used. It looks like that: a computer is infected with a stealth virus. The virus is active in memory, a user runs his scanner and is searching for strings in files-but as it opens files with viruses, it can't find anything because the virus hides itself. What to do with stealth viruses?

Memory scanning

You have to think for both sides otherwise you can't rule this war. The simplest answer is to scan the memory as well and to get rid of the virus.

Memory scanning is similar to file scanning. All memory is checked for the same strings as files, if found - a virus is reported in memory. To speed up the process some antiviruses don't scan the whole memory but only possible locations - ROMs, antivirus itself, etc. Once a virus is found, some antiviruses are able to patch it to be inactive without need to boot from a clean floppy. Inactivating can be done easily by replacing virus handlers with jump to original entry-point of hooked interrupt. Also a virus body is erased not to report the virus again.

Another idea how to partially inactivate a virus in memory presented by some antiviruses is a known entry-point method. If you know the original entry point you may find out if some virus is in memory and you can access functions without a virus's influence.

Encryption

It started with the first encrypted viruses that had main body encrypted. But there had to be at least one short decryption routine. And this routine is a fixed sequence of bytes, and can be identified with a scan-string.

A problem that arises here is encryption vs. body dilemma. Identifying a virus by encryptor only, it can't make a difference between various viruses with the same decryptor. The cleaning problem can be solved by cleaning routine if you want to get rid of encrypted virus. But this is an important question in methodology and it still remains – there is no exact identification before cleaning and identification must be done again during a cleaning process in different conditions.

11. Securing Computer Systems

Securing Computer Systems

You have probably heard a friend or neighbour complain that computers are "taking over society" and that they "know too much about us". Whether you agree with this point of view or not, serious computer professionals recognize that current information processing trends, such as the emergence of massive credit databases containing the credit history of 150 million Americans, can potentially jeopardize our personal freedom.

Although there is no reversing the trend toward large credit databases, their existence brings up two important social and organizational issues: privacy and security. Privacy is an individual concern; people need assurance that their personal information, such as employment and credit history, will be used properly. Security is an organizational concern; businesses need safeguards that protect computer systems and data from damage or unlawful use. As a result, security protects hardware, software, and data from natural disasters such as fire, flood, and earthquake, it also guards against sabotage and espionage, as well as various kinds of theft.

Computer crime, which results from the deliberate tampering with data, has become a costly trend for organizations ranging from small businesses to multinational corporations and the federal government. We will focus on methods that companies use to protect against computer crime; such methods will also help to protect data from natural disasters.

One study of 283 businesses and governmental institutions found that half were victims of mputer crime each year. Records show that while a bank robber armed with a gun steals an av- ıge of $1,600 from a bank, the white-collar criminal armed with a computer steals an average of 00,000! These statistics for white-collar crimes may in fact be low, because many organizations ch as banks, fearing a loss of customer confidence, do not report all the crimes that occur.

Computer crime can be as small-scale as the unauthorized use of business computers for ırsonal purposes, or as large as using computers to send corporate financial assets to a pri- ite Swiss bank account. Computer crime can include:

Theft of Computer Time. This common theft may be as simple as students doing term pa- ırs on computers at work or as great as someone stealing thousands of dollars of processing ne for personal profit.

Manipulation of Computer Programs or Data. Remember the movie *War Games,* in which e protagonist changed his grades in the high school computer by using a modem at home? nfortunately, such actions are all too common. Newspapers frequently print stories about ımputer enthusiasts, or "hackers", breaking into public and private databases - sometimes st for the technical challenge of it! Some hackers even plant "computer viruses" that tie up PUs by commanding them to do such time-consuming tasks as attempting to calculate the ecise value of pi - thereby crippling enormous data networks.

Theft of Data and Assets. Legally called embezzlement, computer theft often involves a usted employee who cannot resist the temptation to steal. All too often, the theft is amazingly asy to carry out. For example, the chairman of the board and some of the executive officers f the Equity Funding Corporation - a publicly held mutual fund and insurance company - used ırminals to enter insurance policies for nonexistent people. By entering bogus data over a pe- od of years, they greatly increased the company's apparent assets. Of the 97,000 insurance olicies maintained by the computer, almost two-thirds were fictitious policies with a face value f $2.1 billion! Because the company appeared to be more profitable than it actually was, the tock of Equity Funding was greatly inflated; consequently, the officers, who were large share- olders, were able to sell their shares at considerable profits. The fraud went undetected be- ause auditors consistently accepted the computer printouts that listed policyholders as be- ond question. The crime was discovered by auditors and the police only when a former em- loyee revealed the scheme.

Users Stealing Software. Are you stealing when you make a copy of word processing software or a friend to use? The law says you are. Yet many normally law-abiding people who would not ream of shoplifting see nothing wrong with making a copy of a $500 word processing package and giving it to a friend, despite the fact that it is in violation of the U.S. Copyright Code. Bootleg- ging software is clearly unethical, as well as illegal, yet it is so widespread that estimates suggest here may be as many as ten illegal copies for every legitimate program sold.

To prevent bootlegging, software vendors sometimes copy protect software so that it cannot be copied to other disks - or can only be copied a limited number of times, generally twice. Copy-protected software is becoming rarer; however, as vendors respond to complaints from users that copy protection is a nuisance. Copy protection can make it difficult to do legitimate copying, such as creating backup disks or recovering from hard-disk failure. In fact, the lack of copy protection is often a selling feature for many programs.

Limits of Legislation to Control Computer Crime

Legislators have been trying for years to enact laws to curb computer crime at the federal level, without success. The major stumbling block is the lack of a legal definition for "property" and "value" as they relate to computerized information. Prosecutors are also reluctant to take computer thieves to court because the computer security in an organization is typically so poor and computer crime laws so vague that they find it difficult to build good cases. Because the federal government has not enacted effective laws in this area, many states are beginning to pass stronger legislation that protects companies from computer theft, deliberate destruction of data and so on.

Hardware and Software Controls

Precautions and controls can be employed for protecting hardware and software from illegitimate use. Access to a system and its data should be controlled, and secure backup copies maintained. Most electronic network packages include measures for controlling access to sensitive files. Other ways to reduce the risk of fraud are to rotate personnel in certain jobs and to separate important tasks so that no one person could complete a fraudulent transaction.

Controlled Access to Hardware and Software. If data or program security is important, systems analysts should ensure that unauthorized personnel cannot have physical access to the computer system. The system can be housed in a locked room to protect hardware from theft and to prevent unauthorized people from accessing data or programs. Specific physical and electronic access control techniques include keyboard locks, automatic logs, restricted access to systems and limited after-hour use. Using passwords that change periodically can also help to keep a system secure.

Backup Copies of Data and Programs. Creating backups consists of making extra copies of programs or data. Backups safeguard against the loss of programs or data in computer memory, on hard disks or on diskettes. Data should be backed up each rime it is updated or changed in any way.

Security for Backup Copies. Because backups represent the safety net that will be used if original data or programs are destroyed, the backups must be protected with the same care as the originals. Protection techniques include "keeping them under lock and key" in a fireproof container and storing them in a location several miles from the originals. This protects files from natural disasters as well as computer crime.

Distributed Network Controls. Distributed processing and remote databases pose specific control problems for organizations because they are vulnerable to unauthorized access to data from remote locations. For example, a company trying to discover a competitor's confidential data might tap into the competitor's network and monitor transmissions for their own benefit. Two common controls are used for protecting networks: data encryption and callback systems.

- *Data encryption* involves encoding data so that it cannot be understood unless it is first decoded. To encrypt data, a program uses an algorithm to scramble, or encode, data before it is sent over the network. Another program at the receiving end then decodes the data. If an unauthorized person tapped into the database, the data would not readable.

Because data encryption is cumbersome and expensive, it is generally used only for sensitive data. Access to both the encryption and the decoding programs must be tightly controlled as well.

- *A callback system* ensures that a user calling in by phone is calling from an authorized telephone number. After validating a user's password, the callback system breaks the phone connection and then immediately calls a previously stored telephone number linked to the password in the system's memory. If this is not the number where the person seeking access is located, entry to the system will not be permitted.

Separation and Rotation of Functions. Another way to ensure the security of computer systems is to segment the work done so that each task requires several different people. This separation of duties minimizes the risk of unauthorized and fraudulent use and/or modification of data as well as programs, because two or more perpetrators within the department would be required to commit a fraud. Separating tasks also reduces the risk of errors since one person can check the work of another.

In addition to separating duties, it is advisable to have duties rotated among individuals within each group from time to time. This arrangement is likely to thwart fraudulent collusion among employees. If, for example, a particular data entry operator who enters only payroll data and a programmer who works exclusively on payroll programs intend to work together to commit a computer crime, they may find it difficult to carry it out if their jobs are periodically rotated.

Safeguards such as these help honest people stay honest. Honest computer users often confront ethical challenges as well, and these issues are often not so clearly defined.

Network Security

Computer security and the need to physically protect data are very important. This is true in networking as well, for whenever a computer is connected to a phone line, it is extremely vulnerable to unauthorized people gaining access to the data stored there.

Protecting Networks. Because networks can make computer systems vulnerable to intrusion, security systems exist to protect them. One such system employs user names, account numbers and passwords to differentiate between authorized users and intruders. To make this system work, users must keep their passwords confidential.

Another method of ensuring network security is the callback system. In this system, you begin by connecting to the computer via telephone line and modem. Next, the computer prompts you to enter your user name and password. At this point, the computer terminates the connection. The computer verifies your user name and password, and then calls your computer back at your preregistered phone number to establish the connection for your work session.

Intruders. People who intrude or break into computer systems are sometimes inappropriately called hackers. A true hacker is someone who demonstrates great skill in programming and working with computers; for that reason, we need to make a distinction between the hacker and the intruder, a person who deliberately gains access to a computer system that is the property of another party. Intruders gain access to computer systems by circumventing user name/password security. Some use an automatic telephone dialer to find the numbers of computer systems. When they locate a number, their computer attempts to guess user names and passwords by trying every possible combination of letters and numbers.

To thwart intruders, systems use lockouts - software that only lets you have three tries at entering a user name or password. After three incorrect attempts to enter a correct user name and password, the phone connection is broken. This can certainly slow down the efforts of an intruder. Another security measure requires that authorized users change passwords at assigned intervals.

The latest technology uses keycards, fingerprints, voiceprints and retinal eye scans to secure multiple-user systems. Although these require additional sophisticated hardware and software, many military, government, corporate, and private institutions often feel their information security is worth the extra cost and effort.

There are many other ways to break into computers and networks. To try figuring out the loopholes before intruders do, some computer owners attempt breaking into their own systems. For example, the U.S. Air Force employs experts from Mitre Corporation to look for weaknesses in its computer security.

Encryption. Encryption is another technique of protecting computer communications. Encryption involves putting coding devices at each end of the communication line. Before sending out a message, one computer encodes the text by substituting what appears to be a gibberish character for real letters. At the other end the message is decoded by the receiving computer. This makes it very hard for transmissions to be read by unauthorized people even if they are intercepted.

File Protection. File protection is now common on personal computers, whether they are used by one person or connected in a network. Individual workers can lock personal files so others cannot read them. In addition, entire disks, whether floppies or hard drives, may also be protected.

The Computer Virus. In recent years, computer systems have been plagued with an insidious program called a virus. A virus is a software program that enters computer systems via other programs or through communications networks, then hides itself. It can go unnoticed for long periods as it infects the computer and then causes it to crash.

Viruses are often created by an employee who has been fired and wants to strike back at the company in anger; by young computer "geniuses" who want to demonstrate their computer skill; by programmers with antisocial tendencies or a perverted sense of humour. Many virus attacks have been had over the past few years, and although they can be stopped, there does not seem to be a way to prevent them.

Artificial Immune Systems: A New Computational Intelligence Approach

An *artificial immune system* (AIS) is a type of optimization algorithm inspired by the principles and processes of the vertebrate immune system. The algorithms typically exploit the immune system's characteristics of learning and memory to solve a problem. They are coupled to artificial intelligence and closely related to genetic algorithms.

AIS began in the mid 80s with Farmer, Packard and Perelson's paper on immune networks (1986). However, it was only in the mid 90s that AIS became a subject area in its own right. The first book on Artificial Immune Systems was edited by Dasgupta in 1999. New ideas, such as danger theory and algorithms inspired by the innate immune system, are also now being explored.

Over the past few decades there has been a growing interest in the use of biology as a source of inspiration for solving computational problems. This area of research is often referred to as *biologically inspired computing*. The motivation of this field is primarily to extract useful metaphors from natural biological systems, in order to create effective computational solutions to complex problems in a wide range of domain areas. The more notable developments have been the neural networks inspired by the working of the brain - Artificial Neural Networks (ANNs), and the evolutionary algorithms (EA) inspired by neo-Darwinian theory of evolution.

The immune system is incredibly robust; it is adaptive, inherently distributed, posses powerful pattern recognition, learning and memory capabilities. It is for these reasons (and more) the immune system is attracting such attention.

The field of AIS is, in computer science and engineering terms, relatively young and is growing at a rapid rate. To date, no single text has ever been written that attempts to consolidate all current research and to focus the many ideas into a general framework.

Fundamentals of the immune System

Understanding the human immune system has significantly advanced in the past few decades. The immune system is known to defend our bodies from attack by foreign invaders (such as viruses and bacteria). It has also been shown to be capable of remembering previous encounters with these invaders and this knowledge has been harnessed to provide vaccines for a whole plethora of viruses.

The Immune System in Context with Other Biological Systems

Immune system research is, in its very nature, interdisciplinary. There are many researchers studying the relationship between the immune system and other biological systems, such as the nervous and endocrine systems. Some cognitive scientists are interested in studying the immune system recognition, learning and memory capabilities. Researchers on evolutionary biology are interested in verifying the contribution of the immune system to the evolution of the organism and comparing their timescales. Finally, researchers on ecological systems argue that immune systems share a number of similarities with ecological economic systems in terms of function.

By studying these biological systems and their integration, new insights for the development of novel hybrid systems and also systems that are closer to artificial life and behaviours than the present ones are expected to be provided.

The working of the brain has led to the development of one of the most influential computational intelligence paradigms: artificial neural networks (ANNs). ANNs have been applied to a vast amount of complex problems such as vision, pattern recognition, classification and approximation, to name a few. By regarding the immune system as cognitive, such as the brain, and making a comparison between them, it is possible to trace new parallels between AIS models and ANN and also to have new insights into how to create novel hybrids between them.

An Artificial Immune System for Network Intrusion Detection

An intrusion detection system (IDS) is an automated system for the detection of computer system intrusions. The main goal of an IDS is to detect unauthorized use, misuse and abuse of computer systems by both system insiders and external intruders. This research proposes a novel approach to building a network-based IDS, which is inspired by a human immune system.

Network-based IDSs monitor any number of hosts on a network by scrutinizing the audit trails of multiple hosts. As the first stage of this research, the set of general requirements of building an effective network-based IDS were found. The three design goals to satisfy these requirements, which are being distributed, self-organizing and lightweight, were identified by a careful examination of the literature. Then, an analysis of the complex capabilities of human immune system was performed and the several salient features of human immune systems for network-based intrusion detection were identified. This analysis shows that the human immune system is distributed through its immune network and unique antibody sets. It is self-organizing because of the three evolutionary processes of gene library evolution, negative selection and clonal selection. It is lightweight because of the generality of approximate binding and gene expression, and the efficiency of memory cells. These remarkable features allow the human immune system to clearly fulfill the design goals for network-based IDSs. Furthermore, even though various approaches have been developed and proposed no network-based IDS has satisfied all its requirements.

The main idea of this model is distinguishing self, which is normal, from non-self, which is abnormal. Many sophisticated network intrusions such as sweeps, co-ordinated attacks and the Internet worms are detected by monitoring the anomalies of network traffic patterns. Thus, the artificial immune model is designed for distinguishing normal network activities from abnormal network activities and is expected to detect various network intrusions.

The artificial immune model for network intrusion detection consists of a primary IDS and a secondary IDS. For a human body, at the bone marrow and the thymus, various detector cells, called antibodies, are continuously generated and distributed to secondary lymph nodes, where antibodies reside to monitor living cells. The distributed antibodies monitor all living cells and detect non-self cells, called antigens, invading into the human body. For the artificial immune model, the primary IDS, which is viewed as the bone marrow and thymus, generates numerous detector sets. They describe abnormal patterns of network traffic packets. They are unique and transferred to each local host. Local hosts are viewed as secondary lymph nodes, detectors as antibodies and network intrusions as antigens. At the secondary IDSs, which are local hosts, detectors are used by background processes which monitor whether non-self network traffic patterns are observed from network traffic patterns profiled at the monitored local host. The primary IDS and each secondary IDS have communicators to allow the transfer of information between each other.

For the proposed artificial immune system, the several sophisticated mechanisms of the human immune system which allow it to satisfy three design goals of a competent network-based IDS are embedded in three evolutionary stages: gene library evolution, negative selection and clonal selection. While the currently existing computer immune models focus on the use of a single significant stage according to their perceived purpose, the new artificial immune for this research combines these three significant evolutionary stages into a single methodology.

Gene library evolution simulates the first stage of evolution, which learns knowledge of currently existing antigens. This process allows the model to be lightweight and self-organizing. *Gene expression* and *negative selection* form the second stage of evolution, generating diverse pre-detectors and selecting mature detector sets by eliminating false pre-detectors in a self-organizing way. The transfer of unique detector sets to the secondary IDS also occurs at this stage, making the model distributed. *Clonal selection* is the third stage of evolution, detecting various intrusions with a limited number of detector sets using approximate binding, and generating memory detectors. The generality and efficiency of these mechanisms results in the model being lightweight. In addition, this process drives the gene library evolution in the primary IDS. These three stages are co-ordinated across a network to satisfy the three goals for designing effective IDSs: being distributed, self-organizing and lightweight.

UNIT III. FLEDGING TECHNOLOGIES

The Fifth Generation of Computers

With the fifth generation of computing (1983 - ?) we have personal computers as powerful as mainframes and software that performs certain thinking functions only our brains could do. One sign of fifth-generation activity was the recognition of legitimate expert systems research around 1983. Another was the fifth generation government-funded research project that Japan formally launched in the early 1980s. The goal was to develop computer systems that combine advances in speech recognition, vision systems, database technology, and the telephone booth so that the average citizen would have extraordinarily easy access to computer power and information. The Japanese project is still underway but today, fifth-generation projects are in development around the world - especially in the United States.

Circuit Technology

One fifth-generation goal is to develop VHSIC and UHSIC chips, more commonly called superchips. Early superchips are two to three times faster than the fastest personal computer microprocessor chip. The first uses for these superchips are in military applications, but scientists say they'll be widely used in consumer products such as digital television sets. Not only will superchips be fast, they'll be self-repairing, too. They'll be able to diagnose their own malfunctions and substitute spare transistors on the same chip.

Chip development is truly at the center of the fifth generation. Other research is underway in the following areas:

-New materials such as gallium arsenide to replace silicon.

-New techniques such as superconductivity or operating in extremely low temperatures that speeds the flow of electricity.

-New applications such as the neurocomputer or neural computing chip.

First developed by Drs. Carver Mead and Federico Faggin, the neural chip works very similarly to the neurons in our own brains. The chips can be linked into massive, complex circuits that scientists hope will eventually lead to "thinking" computers.

I/O Devices

Touch screens will be in common use for simple tasks in the fifth generation. Instead of typing, we'll be able to give these computers voice instructions. In time, they will reply vocally, or by more conventional means if we prefer. As the manner in which the computer inputs and outputs changes, so will the user interface - to a point where it will be entirely transparent, as it should be.

Memory and Storage

Memory chips will acquire vast storage capabilities; economies of scale in manufacturing will drop so dramatically that they will be able to hold tremendous amounts of mass storage. Optical storage devices, based on today's CD-ROM and videodisc technologies, will largely replace magnetic disks. Gradually, we will be able to store data on networks as well, so we won't have to pay for our own storage devices at all.

Programming

Object-oriented programming (OOP) and computer-aided software engineering (CASE) tools will be in routine use for developing applications in business. Indeed, many users will be able to utilize OOP for creating personalized applications. This will be a dramatic advance; applications can be designed to one's own personal needs and tastes. No longer will it be necessary to learn how to use a program; the computer will "learn" how to perform a particular task exactly as you want it done. In time, the "programming language" will be our own spoken language, which the computer interprets into its own machine language. The instructions themselves will be programmable, so that computer "crashes" will be a thing of the past: programs will be self-repairing.

Software

The fifth generation will bring us intelligent interfaces that learn to respond to our personal workstyle or lifestyle. For example, the computer knows that you like to wake up in the morning to the morning news headlines, then you like to read your electronic mail. It will alert you when you need to attend a meeting or write a report. As you change between types of data, it will automatically switch between the spreadsheet and database applications. Image processing and voice recognition will be an important part of fifth-generation systems. We will be on-line continuously as the computer searches for and retrieves data we have previously programmed it to find for us.

CPU Speed

No matter how powerful today's computer is, there are still times when we must sit and wait for an operation to be completed. This is because the CPU can only process one instruction at a time, and even though it may be processing at 20, 25, or even 33 million instructions per second, we still end up staring at the screen for what seems like hours.

Fifth-generation computers will overcome the problem of speed in several ways. One is the new generation of reduced instruction set computers, or RISC machines. They take advantage of the traditional von Neumann CPU architecture by utilizing, as the name implies, fewer instructions. Another is the parallel processing architecture that allows many CPUs to share the processing work. Perhaps the most promising is the optical computer, which uses light instead of electricity. Such a computer would be able to process instructions at nearly the speed of light.

What Lies Ahead?

The computer has made advances that are unparalleled by any other invention or achievement of the human race. Yet many people believe its greatest contributions are yet to come. The fifth generation is leading us into an age where the computer will be our sidekick, companion, and extension of our brain. We will not have to spend so much time learning how to use it or manipulating its hardware and software to get what we need from it. In this exciting new generation, the computer will become a genuine productivity tool. As it does so, we will find that the fifth generation is bringing us computer systems capabilities we can scarcely imagine.

Yet are people's attitudes changing and advancing along with improvements in machines such as the computer? We have seen great resistance on the part of people to change, especially with respect to learning how to understand and use a computer. However, whether we are ready or not computers begin making dramatic changes in our lives over the next 30 years.

We will see much of our daily activities automated by computers. Those who hold jobs will be those who make business decisions or those who work in entertainment and the creative arts: business and government leaders, salespeople, artists, composers, sports players. There will be a vast explosion in creativity, but also a huge leisure class. The quality of life will improve for everyone - but at the same time, the world's population will continue to grow at a rapid pace.

Meanwhile, we will enter the sixth, and perhaps the seventh, computer generations. Computers will be ubiquitous, everywhere and invisible. We will issue instructions to walls, sidewalks, autos, appliances. Robots will tend to our personal needs, and all information and entertainment will be delivered to us, automatically, via fiber-optic networks.

Can we effectively deal with all this change? Author-adventurer Laurens Van der Post said, "Life is its own journey, presupposes its own change and movement and one tries to arrest them at one's eternal peril". We must begin learning how to develop our capacity to change, and expanding our mental and human potential now, in order to prepare for this new world. It is ours and we must be its master, not its slave.

DNA Computers

Even as you read this text, computer chip manufacturers are furiously racing to make the next microprocessor that will topple speed records. Sooner or later, though, this competition is bound to hit a wall. Microprocessors made of silicon will eventually reach their limits of speed and miniaturization. Chip makers need a new material to produce faster computing speeds.

"You won't believe where scientists have found the new material they need to build the next generation of microprocessors. Millions of natural supercomputers exist inside living organisms, including your body. DNA (deoxyribonucleic acid) molecules, the material our genes are made of, have the potential to perform calculations many times faster than the world's most powerful human-built computers. DNA might one day be integrated into a computer chip to create a so-called biochip that will push computers even faster. DNA molecules have already been harnessed to perform complex mathematical problems.

While still in their infancy, DNA computers will be capable of storing billions of times more data than your personal computer.

EUVL Chipmaking

Silicon microprocessors have been the heart of the computing world for more than 40 years. In that time, microprocessor manufacturers have crammed more and more electronic devices onto microprocessors. In accordance with Moore's Law, the number of electronic devices put on a microprocessor has doubled every 18 months. Moore's Law is named after Intel founder Gordon Moore, who predicted in 1965 that microprocessors would double in complexity every two years. Many scientists have predicted that Moore's Law will soon reach its end because of the physical limitations of silicon microprocessors.

The current process used to pack more and more transistors onto a chip is called *deep-ultraviolet lithography* (DUVL), which is a photography-like technique that focuses light through lenses to carve circuit patterns on silicon wafers. It was predicted that DUVL would begin to reach its limit around 2005. And at that time, chipmakers would have to look to other technologies to cram more transistors onto silicon to create more powerful chips. Many scientists are already looking at extreme-ultraviolet lithography (EUVL) as a way to extend the life of silicon at least until the end of the decade. EUVL uses mirrors instead of lenses to focus the light, which allows light with shorter wavelengths to accurately focus on the silicon wafer.

DNA and Quantum

Beyond EUVL, researchers have been looking at alternatives to the traditional microprocessor design. Two of the more interesting emerging technologies are DNA computers and quantum computers.

DNA computers can't be found at your local electronics store yet. The technology is still in development, and didn't even exist as a concept a decade ago. In 1994, Leonard Adleman introduced the idea of using DNA to solve complex mathematical problems. Adleman, a computer scientist at the University of Southern California, came to the conclusion that DNA had computational potential after reading the book "Molecular Biology of the Gene", written by James Watson, who co-discovered the structure of DNA in 1953. In fact, DNA is very similar to a computer hard drive in how it stores permanent information about your genes.

Adleman is often called the inventor of DNA computers. His article in a 1994 issue of the journal Science outlined how to use DNA to solve a well-known mathematical problem, called the *directed Hamilton Path problem,* also known as the "travelling salesman" problem. The goal of the problem is to find the shortest route between a number of cities, going through each city only once. As you add more cities to the problem, the problem becomes more difficult. Adleman decided to find the shortest route between seven cities.

You could probably draw this problem out on paper and come to a solution faster than Adleman did using his DNA test-tube computer. Here are the steps taken in the Adleman DNA computer experiment:

1. Strands of DNA represent the seven cities. In genes, genetic coding is represented by the letters A, T, C and G. Some sequence of these four letters represented each city and possible flight path.
2. These molecules are then mixed in a test tube, with some of these DNA strands sticking together. A chain of these strands represents a possible answer.

3. Within a few seconds, all of the possible combinations of DNA strands, which represent answers, are created in the test tube.

4. Adleman eliminates the wrong molecules through chemical reactions, which leave behind only the flight paths that connect all seven cities.

The success of the Adleman DNA computer proves that DNA can be used to calculate complex mathematical problems. However, this early DNA computer is far from challenging silicon-based computers in terms of speed. The Adleman DNA computer created a group of possible answers very quickly, but it took days for Adleman to narrow down the possibilities. Another drawback of his DNA computer is that it requires human assistance. The goal of the DNA computing field is to create a device that can work independent of human involvement.

DNA computers have the potential to take computing to new levels, picking up where Moore's Law leaves off. There are several advantages to using DNA instead of silicon:

- As long as there are cellular organisms, there will be a supply of DNA.
- The large supply of DNA makes it a cheap resource.
- Unlike traditional microprocessors, which are made using toxic materials, DNA biochips can be made cleanly.
- DNA computers are many times smaller than today's computers.

DNA's key advantage is that it will make computers smaller, while at the same time increasing storage capacity, than any computer that has come before. One pound of DNA has the capacity to store more information than all the electronic computers ever built. The computing power of a teardrop-sized DNA computer, using the DNA logic gate, will be more powerful than the world's most powerful supercomputer. More than 10-trillion DNA molecules can fit into an area no larger than 1 cubic centimeter (0.06 inch3). With this small amount of DNA, a computer would be able to hold 10 terabytes (TB) of data and perform 10-trillion calculations at a time. By adding more DNA, more calculations could be performed.

Unlike conventional computers, DNA computers could perform calculations simultaneously. Conventional computers operate linearly, taking on tasks one at a time. It is parallel computing that will allow DNA to solve complex mathematical problems in hours - problems that might take electrical computers hundreds of years to complete.

Today's computers work by manipulating bits that exist in one of two states: 0 or 1. Quantum computers aren't limited to two states; they encode information as quantum bits, or qubits. A qubit can be a 1 or a 0, or it can exist in a superposition that is simultaneously 1 and 0 or somewhere in between. Qubits represent atoms that are working together to serve as computer memory and a microprocessor. Because a quantum computer can contain these multiple states simultaneously, it has the potential to be millions of times more powerful than today's most powerful supercomputers. A 30-qubit quantum computer would equal the processing power of a conventional computer capable of running at 10 teraops, or trillions of operations per second. Today's fastest supercomputers have achieved speeds of about 2 teraops.

Off the Desk

Already we are seeing powerful computers in non-desktop roles. Laptop computers and personal digital assistants (PDAs) have taken computing out of the office. Wearable computers built into our clothing and jewellery will be with us everywhere we go.

By the end of the decade, we could be wearing our computers instead of sitting in front of them.

Our files will follow us while our computer provides constant feedback about our environment. Voice- and handwriting-recognition software will allow us to interface with our computers without using a mouse or a keyboard. Magnetic RAM and other innovations will soon provide our PC with the same instant-on accessibility that our TV and radio have.

One thing is an absolute certainty: the PC will evolve. It will get faster. It will have more capacity. And it will continue to be an integral part of our lives.

SUPPLEMENTARY READING

Artificial Intelligence

Artificial Intelligence (AI) can be defined as the study of methods by which a computer can simulate aspects of human intelligence. One aim of this study is to design a computer that might be able to reason for itself. A more "attainable" objective of work on AI is the development of systems that can work with natural language, meaning the language that we speak and write as distinct from any programmed computer language. Another aspect of AI is the ability of the computer to search knowledge in a database for the best possible reply to a question, because this has strong parallels with the way that we solve problems ourselves.

AI is divided roughly into two schools of thought: Conventional AI and Computational Intelligence (CI).

Conventional AI mostly involves methods now classified as machine learning, characterized by formalism and statistical analysis. This is also known as symbolic AI, logical AI, neat AI and Good Old Fashioned Artificial Intelligence (GOFAI). Its methods include:

- Expert systems: apply reasoning capabilities to reach a conclusion. An expert system can process large amounts of known information and provide conclusions based on them.
- Case based reasoning
- Bayesian networks
- Behaviour based AI: a modular method building AI systems by hand.

Computational Intelligence involves iterative development or learning. Learning is based on empirical data and is associated with non-symbolic AI, scruffy AI and soft computing. Its methods mainly include:

- Neural networks: systems with very strong pattern recognition capabilities.
- Fuzzy systems: techniques for reasoning under uncertainty, have been widely used in modern industrial and consumer product control systems.
- Evolutionary computation: applies biologically inspired concepts such as populations, mutation and survival of the fittest to generate increasingly better solutions to the problem. These methods most notably are divided into evolutionary algorithms (e.g. genetic algorithms) and swarm intelligence (e.g. ant algorithms).

With hybrid intelligent systems attempts are made to combine these two groups. Expert inference rules can be generated through neural network or production rules from statistical learning such as in ACT-R. It is thought that the human brain uses multiple techniques to both formulate and cross-check results. Thus, systems integration is seen as promising and perhaps necessary for true AI.

History

Early in the 17th century, René Descartes envisioned the bodies of animals as complex but reducible machines, thus formulating the mechanistic theory, also known as the "clockwork paradigm". Wilhelm Schickard created the first mechanical digital calculating machine in 1623, followed by machines of Blaise Pascal (1643) and Gottfried Wilhelm von Leibniz (1671), who also invented the binary system. In the 19th century, Charles Babbage and Ada Lovelace worked on programmable mechanical calculating machines.

Bertrand Russell and Alfred North Whitehead published *Principia Mathematica* in 1910-1913, which revolutionized formal logic. In 1931 Kurt Gödel showed that sufficiently powerful consistent formal systems contain true theorems unprovable by any theorem-proving AI that is systematically deriving all possible theorems from the axioms. In 1941 Konrad Zuse built the first working mechanical program-controlled computers. Warren McCulloch and Walter Pitts published *A Logical Calculus of the Ideas Immanent in Nervous Activity* (1943), laying the foundations for neural networks. Norbert Wiener's *Cybernetics or Control and Communication in the Animal and the Machine* popularizes the term "cybernetics".

The 1950s were a period of active efforts in AI. In 1950, Alan Turing introduced the "Turing test" as a way of operationalizing a test of intelligent behaviour. The first working AI programs were written in 1951 to run on the Ferranti Mark I machine of the University of Manchester: a draughts-playing program written by Christopher Strachey and a chess-playing program written by Dietrich Prinz. John McCarthy coined the term "artificial intelligence" at the first conference devoted to the subject, in 1956. He also invented the Lisp programming language. Joseph Weizenbaum built ELIZA, a chatterbot implementing Rogerian psychotherapy. The birthdate of AI is generally considered to be July 1956 at the Dartmouth Conference, where many of these people met and exchanged ideas.

At the same time, John von Neumann, who had been hired by the RAND Corporation, developed the game theory, which would prove invaluable in the progress of AI research.

During the 1960s and 1970s, Joel Moses demonstrated the power of symbolic reasoning for integration problems in the Macsyma program, the first successful knowledge-based program in mathematics. Leonard Uhr and Charles Vossler published "A Pattern Recognition Program That Generates, Evaluates, and Adjusts Its Own Operators" in 1963, which described one of the first machine learning programs that could adaptively acquire and modify features and thereby overcome the limitations of simple perceptrons of Rosenblatt. Marvin Minsky and Seymour Papert published *Perceptrons*, which demonstrated the limits of simple neural nets. Alain Colmerauer developed the Prolog computer language. Ted Shortliffe demonstrated the power of rule-based systems for knowledge representation and inference in medical diagnosis and therapy in what is sometimes called the first expert system. Hans Moravec developed the first computer-controlled vehicle to autonomously negotiate cluttered obstacle courses.

In the 1980s, neural networks became widely used due to the backpropagation algorithm, first described by Paul Werbos in 1974. The team of Ernst Dickmanns built the first robot cars, driving up to 55 mph on empty streets.

1990s & Turn of the Millennium

The 1990s marked major achievements in many areas of AI and demonstrations of various applications. In 1995, one of Dickmanns' robot cars drove more than 1000 miles in traffic at up to 110 mph. Deep Blue, a chess-playing computer, beat Garry Kasparov in a famous six-game match in 1997. DARPA stated that the costs saved by implementing AI methods for scheduling units in the first Persian Gulf War have repaid the US government's entire investment in AI research since the 1950s. Honda built the first prototypes of humanoid robots.

During the 1990s and 2000s AI has become very influenced by probability theory and statistics. Bayesian networks are the focus of this movement, providing links to more rigorous topics in statistics and engineering such as Markov models and Kalman filters, and bridging the divide between "neat" and "scruffy" approaches. The last few years have also seen a big interest in game theory applied to AI decision making. This new school of AI is sometimes called machine learning. After the September 11, 2001 attacks there has been much renewed interest and funding for threat-detection AI systems, including machine vision research and data-mining. However despite the hype, excitement about Bayesian AI is perhaps now fading again as successful Bayesian models have only appeared for tiny statistical tasks (such as finding principal components probabilistically) and appear to be intractable for general perception and decision making.

AI in Business

Banks use artificial intelligence systems to organize operations, invest in stocks, and manage properties. In August 2001, robots beat humans in a simulated financial trading competition. A medical clinic can use artificial intelligence systems to organize bed schedules, make a staff rotation, and to provide medical information. Many practical applications are dependent on artificial neural networks — networks that pattern their organization in mimicry of a brain's neurons, which have been found to excel in pattern recognition. Financial institutions have long used such systems to detect charges or claims outside of the norm, flagging these for human investigation. Neural networks are also being widely deployed in homeland security, speech and text recognition, medical diagnosis (such as in Concept Processing technology in EMR software), data mining, and e-mail spam filtering.

Robots have become common in many industries. They are often given jobs that are considered dangerous to humans. Robots have proved to be effective in jobs that are very repetitive which may lead to mistakes or accidents due to a lapse in concentration, and in other jobs which humans may find degrading. General Motors uses around 16,000 robots for tasks such as painting, welding, and assembly. Japan is the leader in using robots in the world. In 1995, 700,000 robots were in use worldwide; over 500,000 of which were from Japan.

AI in Fiction

In science fiction AI - almost always strong AI - is commonly portrayed as an upcoming power trying to overthrow human authority as in HAL 9000, Skynet, Colossus and The Matrix or as service humanoids like C-3PO, Marvin, Data, KITT and KARR, the Bicentennial Man, the *Mechas* in A.I., Cortana from the Halo series or Sonny in I, Robot.

A notable exception is Mike in Robert A. Heinlein's *The Moon Is a Harsh Mistress*: a supercomputer that becomes aware and aids in a local revolution.

The inevitability of world domination by out-of-control AI is also argued by some fiction writers like Kevin Warwick. In works such as the Japanese manga *Ghost in the Shell*, the existence of intelligent machines questions the definition of life as organisms rather than a broader category of autonomous entities, establishing a notional concept of systemic intelligence. Some fiction writers, such as Vemor Vinge and Ray Kurzweil, have also speculated that the advent of strong AI is likely to cause abrupt and dramatic societal change. The period of abrupt change is sometimes referred to as "the Singularity".

Author Frank Herbert explored the idea of a time when mankind might ban clever machines entirely. His Dune series makes mention of a rebellion called the Butlerian Jihad in which mankind defeats the smart machines of the future and then imposes a death penalty against any who would again create thinking machines. Often quoted from the fictional Orange Catholic Bible, "Thou shall not make a machine in the likeness of a human mind."

A Robot's Best Friend

Tired of walking your dog and finding its hair everywhere? Sony's robotic dog, AIBO, may be for you.

AIBO is a totally autonomous robot capable of hearing and seeing, sensing balance and touch. Eighteen specialized motors allow such dog-like motions as rolling over, scratching, playing dead, and chasing a pink ball. Like a puppy, with time and training AIBO develops perfect movements and unique behaviour patterns. Programmed to seek companionship, AIBO simulates emotions like happiness, surprise, and anger and is clever at responding to verbal commands.

Computerized Canine That Is Full of Byte

A robot dog that is said to be the world's largest and most technologically sophisticated canine pet has been let off its leash by its designers in the United Kingdom.

The RS-01 is similar to a real dog in that it can understand and act on up to 60 verbal instructions, thanks to voice-recognition technology. It can walk, perform tricks, sit up and beg, respond to being stroked and even recognize different coloured balls.

The RS-01 also has practical uses, such as allowing its owner to view locations remotely via an onboard camera, and to read e-mails. Acting as a mobile closed-circuit surveillance system, it can also play the role of a guard dog, watching out for intruders.

The computerized canine was designed by experts at RoboScience, a company in England that specializes in robotic technology. The cyber-pet is 82 centimetres (cm) long, 67cm tall and 37cm wide (the size of a Labrador) and is much bigger than any other robot dog produced so far.

A limited number of handmade RS-O1s will be produced to order, depending on the demand and will be available at between 15,000 and 20,000 pounds sterling each. The design team is led by chief technical officer Nick Wirth.

The RS-01 is packed with cutting-edge technology. It can learn tricks, track you down in your house and has sensors that can recognize different colours. If you pat or touch its paw, it will move.

The only things that are anywhere near as sophisticated are prototypes costing over a million pounds each. RoboScience achieved this in just seven months with a team of eight people and six figure budget.

The technology had applications in the field of space exploration, the leisure industry, special effects, security and the military.

Thinking Robots

Have you ever heard anyone yell "Think fast!" and then toss something at you? It's a dirty trick, but the amazing part is that you can usually manage to dodge, block, or catch the item, as you see fit. Your brain is fast, it can make judgements "on the fly" and adapt to changing situations.

Traditionally, efforts to mimic human thought have centered around rule-based logic. The computer in front of you now uses rule-based logic: binary data is stored and manipulated according to a set of pre-programmed rules. Most robotic "brains" are rule-based, often contained on a single chip that functions as a computer - a microcomputer.

Rule-based systems can be used to create artificial intelligence, by programming vast amounts of information into a computer. Relying on this enormous set of data, such a computer is able to mimic intelligence, for example, helping to diagnose diseases by comparing symptoms to those in a database. Such "expert systems" can know more than any single person, and yet they have only a very narrow range of useful function. Also, they can't learn. They can only make connections they've been programmed to make.

Another approach to artificial intelligence is neural networks. Neural networks are modelled after the human brain, with the advantage that they are better at handling ambiguity than rule-based systems. A neural net "learns" by exposure to lots of inputs and corresponding outputs. Once trained, the neural net responds to an input with a likely output.

Unlike rule-based systems, a neural network doesn't give definite answers, only most probable answers (some call this "fuzzy logic"). Sounds wishy-washy, perhaps, but many real problems - for example, "Will it rain today?" - don't have definite answers.

A third and relatively new approach to robotic intelligence is something called a stimulus-response mechanism (also known as subsumption architecture), pioneered by Rodney Brooks at MIT. In a stimulus-response robot, there is no memory and no logical decision-making, only hard-wired responses to stimulation.

For example, by linking light sensors directly to motors, it's possible to make a light-seeking robot. Several stimulus-response mechanisms operating simultaneously in one robot can create elaborate behaviour that seems intelligent.

Mushroom Pickers Wanted: Only Robots Need Apply

Think of it - not a human being in site - yet mushrooms are being picked and greens on golf courses are being cut. If work being carried out at Warwick University continues to be successful, those unlikely scenarios could be commonplace in the United Kingdom by 2016.

Yes, robots are on the march again into one of the last bastions of labour-intensive industry - farming and horticulture.

Research engineers and horticulture specialists at Warwick University, central England, are working together to devise a suite of robots and automated systems that could transform farming and horticulture over the next decade.

The researchers from Warwick's horticultural arm, and its manufacturing engineering section, are working on a number of products that will vastly reduce the labour costs of farmers and growers. These include:

- *Robotic mushroom picker:* the robot uses a *charged-coupled camera to spot and select* only mushrooms of the exact size required for picking - achieving levels of accuracy far in excess of human labour. The mushrooms are picked by a suction cup on the end of a robotic arm. Although the picking speed is just over half that of a human, the robot can be set to pick 24 hours a day right through the night without any break. The researchers also hope to increase the picking speed to much closer to that of a human.

- Inflatable conveyor belt: a revolutionary group of inflatable aids to harvesting that provide huge savings on labour costs. The inflatable conveyor system can be driven into an open field or covered growing area. Within minutes, up to 100 metres of powered conveyor belt can be deployed allowing crops to be processed at high speed straight to cool storage, or washing, or simply sorted and graded while still in the field.

- Robot grass cutter: mowing the lawn is a drudge but for growers, farmers, even golf course owners, with large amounts of grassland it is a massive problem with every tractor requiring a skilled employee to manage such pastures. Warwick Manufacturing Group researchers are developing a new method that can allow a farmer or grower to deploy multiple robotic grass-cutting machines at the same time all under the supervision of just a single employee. They are working with the Ransomes Spider grass-cutting device that can be remotely controlled and can mow on 40-degree inclines. They are replacing that remote control with a computer that can use its own data sensors attached to the mower, to travel autonomously across fields, working in groups with other robotic mowers, to ensure that the field is mowed as quickly as possible.

The climate of the UK determines the pattern of agriculture. Despite being over 50 degrees north in latitude the environment is temperate, influenced by the warm Gulf Stream. The geology of the UK is very varied and the result of these two criteria has been the development of a diverse and fertile agriculture with a long tradition of quality crops and livestock.

UK agro-food production and processing technologies are among the most advanced in the world. The industries that support them have also developed to an advanced state and the sector as a whole has both the expertise and the technology to satisfy the most exacting needs of overseas markets.

Britain is a country whose widely diverse landscapes have sustained and nurtured strong traditions in agriculture. Furthermore, Britain is a country that is pioneering new technology, crops and stock to improve the quality, efficiency and safety of agriculture and the food it produces.

The natural advantages of fertile soils and a favourable climate, careful breeding of plants and livestock, scientific management of production systems, crop harvesting and storage - together these have nurtured a rich and diverse agriculture second to none in the world.

The success of British agriculture would have been impossible without the appropriate technology. As the birthplace of the industrial revolution, Britain has long been world famous, for quality and innovation in engineering. British agriculture has benefited directly from this expertise and benefits directly from manufacturers producing a wide range of world-class agricultural machinery and equipment.

In recent years the application of economic, environmental and consumer pressures has greatly influenced the development of the agriculture and food industries in the UK. As a result, British agro-food production and processing technologies have developed to an advanced state.

Crossword Wizard

This machine thinks it's clever.

IS NOTHING sacred? First computers learnt how to defeat people at chess and backgammon, and now, thanks to Duke University in Durham, North Carolina, they can even take us on at solving crossword puzzles—and sometimes win.

In March, Michael Littman of Duke University's computer science department, took his crossword-solving software, Proverb, to the American Crossword Puzzle Tournament in Stamford, Connecticut. While Proverb didn't compete in the contest, Littman reports that after analyzing its performance in solving the same 15-minute puzzles presented to the contestants, it would have been placed 147th out of the 254 human solvers. These results were presented to a conference of the American Association of Artificial Intelligence in Orlando, Florida.

Proverb, which runs on several PCs simultaneously, includes a complex array of databases and query algorithms, written in four different programming languages. It tackles puzzles in a way completely alien to human solvers. Faced with a typical quick newspaper puzzle, with clues such as "Sicilian spewer" (answer: ETNA), Proverb starts by sending all the clues and word lengths to each of its 30 modules, most of which are databases.

Each software module then returns with a list - from dozens to over a thousand - of possible answers, ranking them according to the probability that each is correct. One database comprises 400 000 clues and associated answers from newspaper crosswords over the past 14 years.

Another module is what Littman calls a "degenerate" database which only looks for answers to clues based on direction: if words such as "point", "compass" or "direction" appear in clues for 3-letter words, the database will return answers like NNW and NNE— favourites of some puzzle compilers. One of the largest modules Proverb taps into is the ever-expanding Internet Movie Data- base, which is so big that Littman's team wrote a dedicated search engine to prevent searches taking too long and slowing down the overall crossword puzzle solving process.

Proverb was written in four computer languages because each has its own benefits, depending upon the task at hand. Simple searches to match crossword clues were written in PERL as it's good at scanning arbitrary text files and extracting information from them. The C language was used for number-crunching probabilities to decide on the best answer. Stringing the ensemble together were programs written in C++ and Java that coordinated the various modules.

Proverb takes the top-ranked trial answers - lists from each database module - and puts another program, "Merger", to work. "Merger" balances the results from the lists, assigning a single probability to a master list of possible answers. Finally, a program called Solver takes Merger's output and plugs the highest-scoring words into the crossword grid, then repeats the process, looking for the best fit until it runs out of time.

The idea for Proverb was hatched in 1997 when Will Shortz, puzzle editor at The New York Times, stated - shortly after IBM's Deep Blue computer defeated Garry Kasparov at chess - that computers would never understand English well enough to match humans. Of course, the computer does not really understand English. All it can do is to find statistical relationships between words. But the [crossword] tournament showed that this is good enough to solve crosswords at a competitive level.

Computers as Mind Readers

The marriage of man (or woman) and machine is one of the most intriguing images in science fiction. From the Bionic Woman to RoboCop, these creatures are blessed with bodies that just won't quit and brains at the top of the evolutionary scale. You have seen nothing yet. To some futurists, the most alluring possibility is what science fiction calls "wetware"; the linking of the human brain and computers. The word "wet" refers to the brain; its play on hardware (computer equipment) and software (computer programs). In this vision humans would be connected directly to the machines. The computer could literally read your brain waves, your thoughts, mundane and majestic.

Need a phone number for a friend name Joe? There it is on the screen, called up from your private database, which also lists Joe's other vital stats. That novel in your head? It is all typed out for you on the screen right down to that embarrassing little fantasy you'd prefer no one knew about. (The name of this genre? What else but stream of consciousness?)

Virtual telepathy is probably generations away (if it ever happens), but researchers are currently experimenting with devices that might someday evolve into a kind of wetware. Scientists are trying to create computer images through electrodes attached to the brain, arm or facial muscles. These systems work by translating the electrical signals generated by the nervous system into patterns that the computer can read. The research helps increase computer access for disabled people who could substitute a blink of an eye or the twitch of a cheek for fingers on the keyboard.

A handful of entrepreneurs are working on ways of using this technology to sell products to a larger audience. None of these is real wetware - or even close to it. But they're appealing to true wetware believers.

The *BioMuse* computer from BioControl Systems, a company in Palo Alto, California, processes signals from muscles, eyes and brains, according to cofounder Anthony Lloyd. His partners are an engineer and a neurophysiologist. An armband or sweatband picks up the electrical signals. The BioMuse isn't reading the user's mind in the science-fiction sense; it turns the body's electrical impulses into digital data that the computer understands. Ultimately, the company hopes, the BioMuse could allow users to control the computer through thinking. The computer could interpret thought patterns as different commands, depending on the software that's used with it.

Another small company, IBVA Technologies, Inc., in New York, says its Interactive *Brainwave Visual Analyzer*, transforms brain waves, again collected through a device that looks like a headband, into many forms, including music. As you become angry, for example, your brain waves change and the notes corresponding to particular patterns shift as well. It's a cousin of biofeedback, says Helen Meschkow, IBVA's sales manager. She says in future incarnations, the machine might be used to turn your whole house into a kind of mood ring. If you come home feeling stressed, the machine would translate that tension into a command to lower the lights and turn on soothing music.

MindSet was developed by the AquaThought Foundation, a California research organization dedicated to studying interaction between humans and dolphins, and Monsoon Software of Baltimore. It draws maps of a user's brain waves on a computer screen. Sunil Gupta of Monsoon Software, who helped to create MindSet, says it sells for around $2,000. He thinks devices like his could eventually be used to control computer functions. Someday, Gupta predicts, interaction between humans and computers will be "transparent" - in other words, there will be no artificial barriers such as a keyboard. Machines and humans will interact using a range of senses - auditory, visual and tactile.

Brain waves: In March, Advanced Neurotechnologies in Colorado Springs announced its *Brain Link* computer-interface system. After a two-week, $6,000 training session, users can learn to control their brain waves, claims founder Richard Patton. The user studies a pattern on the screen and then concentrates so he can reproduce that pattern. If some of this stuff sounds on the edge, consider the subculture of neurohackers described by writer Garcth Branwyn in an article on wetware research in *Wired magazine*. "Science fiction has fed us so many images of technologically souped-up humans that the current work ... seems almost retro by comparison," says Branwyn. The neurohackers can't wait for the future he says; they've decided to "take matters into their own heads" sometimes by sending electrical signals directly to their brain through homemade devices.

The early forms of wetware seem especially primitive compared with the brain itself, which is far more complicated than even the most powerful supercomputer man could attach to it today. "We have this gigantic power of memorizing visual images," says Emilio Bizzi, head of MIT's department of brain and cognitive sciences. "No machine could come close to storing a fraction of the images in our head". On the other hand, someday a version of wetware might remind you where you left your car keys so you can concentrate on more important issues - like where you want to go.

Voice - powered Games

Ever find yourself shouting at a computer game? Then you might enjoy playing some that let you use your voice instead of a normal thumb-twiddling controller. Middlesex University, London, UK, has developed two cool games that players can control using their voices. Sing

Pong is a version of the classic game Pong where each player controls a paddle using the loudness of their voice. "sssSnake" looks funny. It involves two players and a square table with a microphone on each side. Images of a snake and a coin are projected onto the table from above, and controlled by each player. The one controlling the snake moves it by making "SSS" noises into different microphones, while the one operating the coin moves it by making "AAHHH" noises. The person controlling the snake must try to eat the coin and the person in charge of the coin has to try to get away.

How to Write On-screen without a Keyboard

The eyes have it, literally ... disabled computer users can now write on-screen using just their eyes, thanks to a tremendous high-tech breakthrough.

This is as a result of innovative software called Dasher, the development of which has been supported by two charities, the Entertainment Software Charity and the ACE Centre that provide or support communication and educational technology for young people with difficulties.

The word-prediction software is operated on a special gaze machine and has been tested by disabled people unable to use a mouse or keyboard. With this software they can "type" just as quickly and efficiently by moving their eyes.

The eye-tracking version of Dasher allows an experienced user to write text as fast as normal handwriting. The system pan be used to write efficiently in any language and the software is free.

The text entry system, Dasher, designed by Dr. MacKay and colleague David Ward at Cambridge University can be controlled by an eye-tracker, a camera that tracks where on the screen the user is looking.

"The software works like a video game in which the user steers ever deeper into an enormous library", explained Dr. MacKay. "A language model is used to shape this library in such a way that it is quick and easy to select probable sequences of characters and hard to make spelling mistakes".

The system is much faster than any alternative writing systems driven by eye-tracker. Experiments show that with practice, Dasher can produce up to 25 words-per minute. Users writing with other eye-tracker-based techniques, using on-screen keyboards, can produce only 15 words per minute.

"Not only is this faster than any alternative writing system driven by an eye-tracker [but] the frequency of spelling mistakes is about five times smaller - and the new system is also less stressful to use," added Dr. MacKay.

Dasher is distinctive because it is controlled by continuous pointing gestures; therefore it uses humans' natural ability to make high-precision analogue movements. A keyboard that is operated with hands using all-or-nothing movements to produce text wastes this ability.

The system also makes no distinction between word completion and ordinary writing. Other writing systems will suggest completed words in a separate part of the display and the user has to point or stare at them to choose them.

Dasher's suggestions are integrated seamlessly into the writing process. The language model adapts to an individual user's writing style, allowing - sometimes - several words to be written with a single glance.

The system will be particularly useful for computer users who are unable to type using a conventional keyboard. Because it can be driven using any pointing device – mouse, rollerball, touch pad or eye-tracker - it is also useful for handheld computers or mobile phones that have no space for a keyboard. It also has potential as an input system for other languages, such as Japanese.

Dasher is a zooming interface. You point where you want to go and the display zooms in wherever you point. The world into which you are zooming is painted with letters so that any point you zoom in on corresponds to a piece of text. The more you zoom in, the longer the piece of text you have written. You choose what you write by choosing where to zoom.

To make the interface efficient, Dasher uses the predictions of a language model to determine how much of the world is devoted to each piece of text. Probable pieces of text are given more space so they are quick and easy to select. Improbable pieces of text (for example, text with spelling mistakes) are given less space so they are harder to write. The language model learns all the time: if you use a new word once, it is easier to write next time.

A big advantage of Dasher over other predictive text-entry interfaces that offer word completions to the user is that it is mode-free: the user does not need to switch from a writing mode to an "accept model predictions" mode. Another advantage is that it is easy to train the model on any writing style - simply load up an example file, then write away.

Founded in 2000 the Entertainment Software Charity (ESC) was created to use the popularity of gaming to help those less fortunate. The not-for-profit charity was established by prominent members of the gaming community to create a dedicated initiative program to advance young people through play, training and education.

The ESC works with hardware vendors, developers, software publishers, magazine publishers, retailers, distributors and logistics providers, all of whom give their time, input, resources and intellectual properties to help support charities committed to improving the future of young people.

The Oxford ACE Centre provides a focus for the use of technology with the communication and educational needs of young people with physical and communication difficulties. It offers a variety of services including in-depth individual assessments, information, R&D and specialist training for parents and professionals.

ACE and Cambridge University worked together on the Dasher Eye Control Optimization project and are now part of a prestigious pan-European project integrating eye gaze for the benefit of users with disabilities, focusing on motor-control disorders.

ACE has no particular company or product interests to promote, including its own software and publications. When it evaluates a service or product it does so purely on the basis of professional opinion and the product's merits. The staff comprises teachers, speech therapists, occupational therapists and technical experts with extensive experience in working with the application of technology to alternative and augmentative communication.

The ACE Centre, founded in 1984, is a registered charity, partly funded by the Department for Education & Employment via the British Educational Communications & Technology Agency, and the Gatsby Charitable Foundation.

Warning on Search Engine Safety

Some net searches are leading users to websites that expose them to spam, spyware and other dangerous downloads, reveals a report.

According to the research the most dangerous words to search for are "free screensavers". The report found that 64% of the sites found using this phrase were flagged as causing problems for users. The authors urged search sites to tighten up rules to ensure users are not inadvertently exposed to harm.

It is well known that visiting sites offering porn, gambling and free MP3s leaves users at serious risk of falling victim to spyware and adware. However, the research by Ben Edelman and Hannah Rosenbaum reveals that those carrying out searches for innocuous subjects are at risk too.

The report looked at the websites returned for 1,394 popular keywords searches found via Google, Yahoo, MSN, AOL and Ask.

The results returned for each search term were then analyzed using the Site Advisor security tool. Once installed this piece of software warns users when they browse websites known to be dangerous.

The most benign of the pages that Site Advisor flags up try to change browser settings (to redirect people to ad sites) and the most dangerous deluge users with spam or bundle adware and spyware in with downloads. In one case signing up with one site led to a test e-mail address getting more than 300 spam messages per week.

DANGEROUS KEYWORDS: Free screensavers, BearShare, Screensavers, Winmx, Limewire, Download Yahoo messenger, Lime wire, Free ringtones.

Some of these risky sites use security flaws and loopholes in browsers to install software without users' knowledge and can lead to that machine being hijacked or to a user losing personal data. The riskiest search terms were associated with downloads (such as "screensavers" and "free ringtones") and file-sharing (such as "BearShare" and "limewire"). Searching under these categories returned a substantial proportion of dangerous sites.

The authors speculate that spammers and scammers are turning to websites to try to snare victims as efforts are made to stop spam before it reaches e-mail inboxes.

"Where internet users go, attackers follow," wrote the authors.

Across all searches approximately 4-6% of sites returned were flagged as dangerous. The authors noted that this was more "alarming" than it first appeared because American net users carry out almost 6 billion searches per month. This translates to 285 million clicks on these potentially dangerous sites every month.

"Even a single visit to a dangerous site can have serious and lasting implications for the average internet user," wrote the authors.

The number of risky sites increases when users click on sponsored results - the adverts generated to accompany particular search terms. Dangerous sites are two to four times as common in sponsored results found the research. "We are troubled by the untrustworthiness of search engines' ads," said the authors.

The authors urged the search engines to get much tougher on those who buy adverts to accompany searches and expose those that abuse visitors.

"We're alarmed by the scope of these problems - by the many ways search engines lead users to sites that turn out to be untrustworthy or worse," concluded the report.

VOCABULARY

A

abuse	1) оскорбление; 2) злоупотребление 3) неправильное использование
accuracy	точность
acquire	приобретать
actuator	силовой привод, рукоятка привода
ad hoc	специальный, устроенный для определенной цели
adapt	приспособлять(ся)
adjust	1) приспосабливать; 2) регулировать, устанавливать
adolescent	1) юный, подростковый; 2) юноша, девушка, подросток
adversarial	различный; противоположный
affective	эмоциональный
alert	предупреждать
alias	псевдоним, альтернативное имя
aliasing	диагональная или кривая линия
ambient	окружающий, обтекающий
ambiguity	неопределенность, неясность
amplification	1) увеличение; расширение; 2) развитие; разработка
anger	гнев, злость
application	применение; приложение
approximate	1) приближаться; почти соответствовать; 2) приблизительно равняться
arouse	пробуждать; вызывать, возбуждать
attachment	(при)соединение; приспособление
attract	привлекать, притягивать
audible	слышный, внятный, слышимый
augment	увеличивать, прибавлять
authentic	достоверный; аутентичный
availability	доступность
awkward	неудобный

B

backplate	сигнальная пластина
bandwidth	полоса пропускания, пропускная способность
baubles	безделушки
Bayesian networks	байесовские сети (используя байесовские сети, программы могут динамически обучаться, постоянно модифицируя вероятности при фиксированном наборе правил)
behaviour	поведение
benchmark	1) отметка уровня; отметка высоты; 2) исходный пункт
Bluetooth	технология беспроводной ближней микроволновой связи

blurry	неясный, туманный, расплывчатый
bone marrow	костный мозг
boot sector	загрузочный сектор
bootlegging	незаконное распространение
boredom	тоска, скука
bounce	отскакивать рикошетом от чего-то; подпрыгивать
breakthrough	крупное достижение, открытие, прорыв
breathe	дышать
brittle	хрупкий, ломкий
brute	неразумный, бессмысленный
bud off	заселять, размножаться

C

calibrate	калибровать, градуировать; проверять, обследовать, выверять
callback	перезванивание, обратный вызов
calm	спокойный, тихий
capture,v	улавливать
cardiac	сердечный
character recognition	распознавание символов
checkmate	нанести поражение
circuit	цепь, контур; схема; сеть
clammy	холодный и влажный
clue	ключ к разгадке; ход мыслей
cluster	1) пучок; 2) скопление, концентрация; 3) совокупность, группа
cognition	познавательная способность
cognitive element	элемент познания
cognitive	познавательный
coherent	ясный, определенный
collision	столкновение
commonplace	общеизвестная истина
compiler	компилятор, программа или тех. устр-во, выполняющее компиляцию
conceal	скрывать
concede	допускать, признавать
concern	забота; проблема
concerted	согласованный
conductivity	*физ.* удельная проводимость
confuse	приводить в замешательство, смущать
consciousness	сознание
consistent	1) совместимый, согласующийся; 2) последовательный
constituent	составная часть
consume	потреблять; расходовать
contamination	заражение
continuous speech	слитная, связная речь
continuous	непрерывный, постоянный
contract	сокращаться

controversial	спорный, дискуссионный
conventional	обычный, общепринятый
conversion	преобразование
convey	1) передавать; 2) выражать
convincing	убедительный
convolutional	свернутый; изогнутый
counterpart	1) двойник; 2) точная копия
creativity	творчество, создание
cripple	наносить ущерб, приводить в негодность
crisp	четкий, ясный
crypting engine	механизм, процесс шифрования
cue	сигнал
curve	1) кривая *(линия или поверхность)*, дуга; 2) изгиб, закругление

D

debugging	отладка
decline	1) уменьшаться; 2) ухудшаться
dedicate	посвящать
define	определять
delete	удалять
deliberate	продуманный; намеренный
detect	выявлять, находить, обнаруживать
detector set	последовательность обнаружения
determine	определять
digest	усваивать
disambiguate	определить, выяснить
disclosing	очевидный
discrete	раздельный, прерывистый, дискретный
discriminate	отличать, различать
disguise	скрыть
disruptive	разрушительный
distil	1) очищать; 2) извлекать сущность
distinct	отдельный; особый, индивидуальный
distinguish	различать
distract	отвлекать
divulge	разглашать
DNA	ДНК
domain	сфера, область
download	загружать
drill	1) сверло, буравчик; 2) дрель
dubious	сомнительный

E

eager	сильно желающий, стремящийся
electric motor	электромотор
elicit	выявлять, устанавливать; извлекать
embed	ввести, внедрить
embody	внедрять, включать
emerge	появляться, выходить; всплывать

emit	издавать *(звук и т.п.)*
empathy	сочувствие; сопереживание
employer	наниматель, работодатель
emulate	1) соревноваться, соперничать; 2) сопоставить
encounter	наталкиваться *(на трудности и т. п.)*
encumber	мешать, затруднять, препятствовать
enhance	увеличивать
enrollment training	обучение приему, восприятию
enticing	привлекательный
entry-point	точка ввода, момент поступления
epistemology	*филос.* эпистемология; теория познания
equation	*мат.* уравнение
estimation	подсчет, вычисления; значение; оценка
evaluate	оценивать
exaggerate	преувеличивать
explication	объяснение; толкование
explicit	точный, определенный
exploit	пользоваться, использовать
exponential	экспоненциальный, показательный
exposure	подвергание какому-то воздействию
extend	расширять
extension	расширение; развитие
extract	извлекать, улавливать
extraneous	внешний, поступающий извне

F

facet	аспект
facial recognition	распознавание лицевых черт, выражения лица
facial	лицевой
facilitate	облегчать
faculty	способность
fan out	распространяться
feasible	осуществимый, выполнимый
feature extraction	выделение признаков
feedback	обратная связь, отклик, реакция
feedforward	идущий вперед; поступательный
feelers	щупальце
fibre – optic network	волоконно-оптические сети
flawless	без изъяна, безупречный, безукоризненный
flick	смахнуть, стряхнуть
flight path	траектория полета
floppy disk	гибкий диск
force	внедрять
foster	поощрять; благоприятствовать
fraud	обман, мошенничество
freeware	свободно (бесплатно) распространяемое ПО
frustrate	расстраивать
frustration	разочарование, безысходность
fusion	слияние, объединение
futility	тщетность
fuzzy	неясный, неопределенный

G

gadgetry	новинки техники
gaze	взгляд
gear	направлять
gene library	клонотека (библиотека) генов, генотека
gibberish	непонятный
glean	тщательно отбирать
glow	оживиться; волноваться
grace	1) украшать; 2) удостаивать
grasp	сжимать; сдавливать
grasp	схватывать; зажимать
grid	решетка, сетка
grind	1) размалывать, перемалывать в порошок; 2) разжевывать
grip	схватить; сжать; захватить
guilt	вина, виновность

H

handwriting recognition	распознавание рукописного текста
hard disk	жесткий диск
harness	использовать
havoc	разрушать опустошать
hazardous	рискованный, опасный
heuristics	*филос.*эвристика (совокупность логических приемов и методических правил теорет. исследования и отыскания истины
hind legs	задняя конечность
	1) отверстие, перфорация; 2) ошибка, просчет
home in	возвращаться, ориентироваться, идти к цели
hopping legs	прыгательная нога
host	ведущий узел; главный компьютер
human expertise	человеческий опыт

I

image recognition	распознавание изображений
imbue	наполнять
impact	влияние
implement	выполнять; осуществлять
imply	1) подразумевать, предполагать; 2) означать, значить
imprint	отпечатывать, оставлять след, запечатлевать
indispensability	необходимость
infancy	начальная, ранняя стадия развития
infer	заключать, делать заключение, вывод
inference machine	устройство вывода
inherent	1) присущий; неотъемлемый; 2) свойственный
innate	врожденный, природный
insidious	коварный; действующий тайно

instant	**немедленный; мгновенный**
instantaneous	**мгновенный**
insurance carrier	*страховой агент*
integrated circuit	интегральная микросхема
intentionality	намерение, стремление, цель
intercept	останавливать, перехватывать
interfere	1) вмешиваться; 2) препятствовать, мешать
intervene	вмешиваться
intervention	**вмешательство**
intrude	**вторгаться, навязываться**
intrusion	**внедрение, проникновение, вторжение**
intrusive	**назойливый**
issue	**спорный вопрос; проблема**
iterative	**повторяющийся**

J

jealousy	ревность
jeopardize	подвергать опасности, угрожать
jukebox	*музыкальный автомат*
justify	обосновывать, подтверждать

K

keyboard lock	блокировка клавиатуры
knowledge base	база знаний
knowledge engineer	инженер по знаниям, специалист, занимающийся формализацией знаний предметной области *для построения базы знаний экспертной* системы
kooky	помешанный, свихнувшийся

L

ladle	*ковш, черпак*
launch	выпускать, запускать
legitimate	законный, легальный
leverage	усиливать
lightweight	несложный
linear	линейный
locomotion	передвижение
loop	1) петля; кольцо; 2) виток; цикл; 3) узел
loophole	лазейка
lymph node	лимфатический узел

M

mainframe	большая ЭВМ (коллективного пользования); универсальная ЭВМ
malfunction	неисправная работа
manipulate	1) осуществлять управление; 2) умело обращаться; 3) воздействовать, влиять

mayhem	повреждение
maze	лабиринт
mediate	связывать
Moore's Law	закон Гордона Мура
multiple	1) составной, сложный; 2) многочисленный
mundane	земной, мирской, светский
muscle tension	мышечное напряжение
myriad	огромное количество

N

narrative	рассказ, повествование
natural language processing	1) обработка текстов (написанных) на естественных языках; 2) обработка с использованием естественного языка
navigate	1) осуществлять управление; 2) передвигаться
niche	убежище; надлежащее место
node	*филос.* узловой пункт
novel approach	новаторский подход
nudge	перевести из одного состояния в другое, побудить, подтолкнуть

O

obscure	скрытый; неясный
obvious	очевидный, явный
occur	происходить, случаться
off-line	автономный, отключенный от сети
omit	пропускать, не включать
on the go	постоянно, целый день
on-line	постоянный, интерактивный, диалоговый
ontology	*филос.* онтология (учение о бытии)
orb	шар; сфера
outward signs	внешние признаки
overall	1) повсюду; 2) полностью
overlap	1) перекрывать; 2) частично совпадать

P

palm	ладонь
parse	делать грамматический разбор
patch	участок, обрывок
pattern recognition	распознавание образов
perceptron	*киберн.* перцептрон
perseverance	осторожность
persevere	стойко, упорно продолжать
pictorial	яркий, живой
piezoelectric	пьезоэлектрический
piggyback	перемещаться, передвигаться
pinwheel	цевочное колесо
pitch rise	высота тона звука

plane	1) плоскость; 2) грань; 3) проекция
plethora	изобилие
plonk	*разг.* дешевое или плохое вино
pop	бросать, кидать
postal codes	почтовый индекс
posture	поза, положение; осанка
preference	предпочтение
prior	1) прежний; предшествующий; 2) важный, веский
profound	1) глубокий; основательный; 2) полный, абсолютный
propagation	распространение
provide	обеспечивать, снабжать
public utilities	коммунальные сооружения, предприятия
pulleys	шкив, блок, ворот
pulse rate	частота пульса
pursue	1) преследовать; 2) следовать намеченному курсу; 3) рассматривать, расследовать; 4) выполнять

Q

quake	дрожание, дрожь
query	запрашивать
quest	поиски
quit	прекращать; оставлять; завершать

R

rage	ярость
random	сделанный или выбранный наугад, случайный, беспорядочный
randomly	наугад, наобум, наудачу
rapport	1) взаимопонимание; согласие; 2) связь, взаимоотношения
rattle	двигаться, мчаться
readings	показания
reasoning	мышление
recipient	получатель
reckon	подсчитывать, вычислять
recognize	распознать, определить
rectangular	1) прямоугольный; 2) жесткий, негнущийся
recurrent	возвратный
refine	улучшать; совершенствовать, очищать
reflect	отражать
regard	рассматривать; считать
rehearsal	1) повторение; 2) репетиция
relapse	повторение; рецидив *(особ. мед.)*
relevance	уместность
relevant	уместный, относящийся к делу
reluctant	неохотный
rely on	полагаться, доверять

remote control	дистанционное управление
replicate	размножаться
representative	образец; представитель
require	нуждаться; требовать
resolution	разрешающая способность
response	ответ; реакция
resultant	1) получающийся в результате, проистекающий; 2) равнодействующий
retrieve	восстанавливать
rigorous	определенный, точный
robotics	робототехника
robust	крепкий, здоровый, сильный
rook	обман, мошенничество
rules of thumb	эмпирическое или практическое правило

S

salient	характерный, выдающийся
scaled-down	упрощенный
scarce	недостающий
schedule	составлять*(список, опись и т.п.)*
screw driver	сверло
scrutinize	внимательно изучить, исследовать
security	безопасность
self – awareness	самосознание
sensible	разумный
sensor	датчик
sensory system	сенсорная система
sequential	последовательный
servomotor	серводвигатель, следящий электродвигатель
share	акция
sheer	явный, очевидный
shell	оболочка; вспомогательная программа, служащая для облегчения работы с операционной системой
shortage	нехватка, недостаток
shrink	уменьшать(ся), сокращать(ся)
sidekick	закадычный друг
skin conductivity	проводимость кожного покрова
snippets	обрывки *(сведений, знаний и т.п.)*
solely	только, исключительно
span	простираться, охватывать
speaker-dependent	зависимый от диктора (система распознавания речи, требующая предварительного обучения или настройки на речь говорящего)
spectacular	эффектный, зрелищный
speech recognition	распознавание речи
spotwelding	точечная сварка
spring-activated	пружинный, приводимый в действие пружиной
squeeze	хватать, сжимать

startle	испугать; сильно удивить
steering	(рулевое) управление
stepper motor	шаговый двигатель, используется для точного позиционирования устройства под управлением компьютера
stimulus	внешний раздражитель, стимул
stimulus-response	(связь) стимул-реакция
stock market	фондовая биржа
store-and-forward system	система с промежуточным хранением (буферизацией) пакетов
strand	цепь, цепочка
stumble	ложный шаг, ошибка
substitute	заменять
subtle	тонкий, неуловимый
supervise	контролировать
supervised	контролируемый
switch	1) устройство; 2) переключатель

T

tactile	осязательный
tailor	приспосабливать
tailorable	приспособленный; подходящий
tangible	осязаемый, материальный
tangibles	нечто ощутимое, реальное, осязаемое
template	шаблон
temporal	временный, преходящий
3-dimension glasses	3-хмерные стереоскопические очки
thwart	мешать, разрушать
thymus	вилочковая железа, зобная железа, тимус
topple	достигать
torch	1) осветительный прибор; 2) паяльная лампа
toss	подбрасывать
touch screen	сенсорный экран
track	следить; прослеживать
tracker	следящая система; следящее устройство
training set	обучающая последовательность
transparent	очевидный, явный
travelling salesman problem	проблема коммивояжера (прямого, кратчайшего пути)
trigger	побуждать, инициировать, дать начало
trust	доверять
tutor	обучать

U

ubiquitous	вездесущий, повсеместный
ultimately	в конечном счете, в конце концов
ultrasonic waves	УЗ-волна (частотой выше 20 кГц)
unobtrusive	ненавязчивый, неназойливый
user name	имя пользователя
utility	услуга

V

vague	неопределенный
valence	валентность
validate	утверждать
value	ценность; важность; величина
vertebrate	позвоночные
vocal	голосовой
voice recognition	распознавание голоса
voice synthesis	синтез речи
volatile	непостоянный, изменчивый; неуловимый

W

wafer	пластина; диск; плата; подложка
widget	1) элемент управления; элемент GUI; элемент графического интерфейса с пользователем; 2) штучка, безделушка, несущественное украшение
wishy-washy	невыразительный, неубедительный, слабый

Y

yield (to)	1) производить; 2) поддаваться, уступать

Учебное издание

Составители:
Михаил Станиславович Венскович
Людмила Николаевна Шпудейко
Наталья Васильевна Кистень

ARTIFICIAL INTELLIGENCE

СБОРНИК ТЕКСТОВ ПО ЧТЕНИЮ НА АНГЛИЙСКОМ ЯЗЫКЕ

для студентов специальности

1- 40 03 01 Искусственный интеллект

Ответственный за выпуск: Венскович М.С.
Редактор: Строкач Т.В.
Компьютерная верстка: Боровикова Е.А.
Корректор: Никитчик Е.В.

Подписано к печати 19.03.2008 г. Формат 80х64 1/16. Бумага "Снегурочка".
Гарнитура Arial Narrow. Усл. п.л. 4,65. Уч.-изд. л. 5,0. Заказ № 373. Тираж 120 экз.
Отпечатано на ризографе Учреждения образования
"Брестский государственный технический университет".
224017. Брест, ул. Московская, 267.