

*John Favaro*

## **The Vodka was Great (but the Meat was Rotten)**

*John Favaro*

**Viareggio, 2 April 2011**

### **Introduction**

When I announced last year that I would give a lecture entitled “The Crowd: Wisdom or Madness?” the only thing on most people’s minds was “What is that title supposed to mean?” Although it was a pretty cryptic title, it wasn’t a totally hopeless task to figure out what the lecture would be about; after all, the book *The Wisdom of Crowds* was available for purchase, so there was a chance that some of you might have heard of it.

But I really do have to concede that this year’s title must seem truly cryptic to you. And yet, believe it or not, there was a small chance of figuring out what it was about. It is something that is buried in the script (online) of my very first lecture, way back in 1992, nearly twenty years ago:

In fact, the best things to come out of machine translation programs were the jokes. As you can imagine, the hardest things to translate are idiomatic phrases and slogans. They gave the program the following phrase to translate into Russian: “The spirit is willing, but the flesh is weak.” The program came up with: “**The vodka was great, but the meat was rotten.**”

### **Artificial Intelligence Revisited**

For many years I had wanted to give a talk here on automatic computer translation of language. It seemed like the perfect topic, because after all that’s what this association is all about: language. So many of us are professionally involved in languages – several are language teachers, some are translators. But the timing never seemed to be quite right. The progress I saw in the automatic language translation programs was not very convincing. They were slow, clumsy, and expensive – and in the end, not very good. But over the last few years, something has changed. Suddenly automatic translation is not just better than before – it’s a *lot* better. So what happened to make it click? That’s what I wanted to talk about today.

But when I went back to my old lecture of nearly twenty years ago to get the quote for my title, I realized that the story of why language translation has gotten so much better has a lot to do with what has happened in many of the areas I talked about in that lecture, and I decided it would be a good idea to take up the story again where I left it off, way back in 1992. And as we’ll see, as I always do in these lectures when I start predicting, I got a few things right – and a lot of things wrong.

## **Computer Chess**

One of the things I talked about in my lecture of twenty years ago was “computer chess.” Back then, there was a lot of interest in building a computer that could match the best human chess players. Even much earlier, I pointed out, people had been fascinated by chess, and I briefly recounted the story of the so-called Mechanical Turk, a contraption that was exhibited for the first time in 1770:

A couple of hundred years ago there was actually a man touring around Europe with a beautiful robot-like machine with arms that hung over a chess board, that would play an excellent game of chess against all comers. However, it was discovered after some time that there was a man hidden inside of the robot who just happened to play a very good game of chess.

(Interestingly, Amazon.com picked up on the term “Mechanical Turk” again to describe a service they introduced in 2005 – because it was about people helping out machines to finish jobs they couldn’t do well yet, like recognizing images. Equally interesting is that the service is based on the idea of crowds – the topic of last year’s lecture.)

People thought that if a computer could beat a human in chess then, well, we were on our way to building intelligent machines. IBM had set itself a “Grand Challenge” to do just that, and had come up with a computer that it called Deep Thought (named after a computer in the series *The Hitchhiker’s Guide to the Galaxy* by Douglas Adams – and I think you can imagine where Adams might have gotten the name). A few years before my lecture in 1992, Deep Thought had managed to beat a grand master and generated a lot of excitement. But when matched against world champion Garry Kasparov, it was roughed up pretty badly. I reported in my talk that Kasparov had declared afterward:

I can’t visualize living with the knowledge that a computer is stronger than the human mind. I had to challenge Deep Thought for this match, to protect the human race.

But I also reported that many felt that the days of the human race’s protection by Kasparov and others like him were numbered:

Now just about everybody accepts that within 5 or ten years a computer will be built that can beat any human being.

That was in 1992, and now we are in a position to know whether that prediction was accurate. Let’s take up that story and see how it turned out.

## **Deep Blue**

After that humiliating defeat, IBM rolled up its sleeves and got back to work. Deep Thought was consigned to the dustbin and a larger, more powerful successor was designed, named Deep Blue. (They got more patriotic – IBM’s nickname is “Big Blue.”). It took a while to get everything worked out, but on May 11, 1997 – just under five years from the date I made that prediction in my talk – Deep Blue beat Garry Kasparov in a six-game match. A computer had finally beaten the world’s best human player.

Kasparov was furious. He said that he had “sometimes seen deep intelligence and creativity” in the computer’s moves. And what did he conclude from this observation? In a bizarre throwback to the Mechanical Turk chess machine of two centuries earlier, Kasparov alleged that Deep Blue had cheated, by getting help from ... humans. Now, if you think you see a paradox in the best human being in the world thinking that the machine that beat him got help from other human beings, you’re not alone. After all, if he was the best human player on Earth, then what other human player could have possibly helped Deep Blue to beat him? The best player on Mars?

The whole episode ended with Kasparov challenging Deep Blue to a rematch. IBM refused and then dismantled Deep Blue, presumably just to make sure that such a rematch wouldn’t happen. (I didn’t read any reports about whether they found a little man inside when they dismantled it.) They also made a pretty good movie about it, called *Game Over: Kasparov and the Machine*. Like Kasparov, the film also implied that there may have been a conspiracy behind it all. But the conspiracy implied by the film was a different one – a plot to boost the stock price of IBM. Actually, that was a much more reasonable conjecture. IBM never claimed it was doing this solely for the good of mankind. Why not advance the state of technology and get some publicity in the process with these Grand Challenges? In fact, IBM was so happy with all the publicity that it set about finding itself another Grand Challenge to work on.

In the first Grand Challenge, IBM had set out to build a computer that could outperform humans in a game. In the next Grand Challenge, IBM decided to set out to build a computer that could outperform humans in ... another game. And this brings us back to the topic of my lecture by way of a television game show that I used to love to watch as a kid.

### **Television Game Shows**

The 1950s and 1960s were a time of fabulous creativity and growth in TV game shows in America, and they were being churned out at an amazing rate during those years. Many of them were eventually syndicated around the world, including Italy. One show from 1956 called *The Price is Right* had a great run in Italy from 1983 to 2001 as *Ok, il prezzo è giusto*, hosted much of that time by Iva Zanicchi (who called the show “the triumph of consumerism”). Another show called *What’s My Line* was invented in 1950, at the very dawn of the television era – and is alive and well today here in Italy, over 60 years later, under the name of *I Soliti Ignoti*, hosted by Fabrizio Frizzi.

The king of daytime television during much of that golden era of game shows was the multi-talented Merv Griffin. Born in 1925 in San Mateo, California (not far from my own birthplace), he had been a child prodigy pianist and soon ended up in show business in Hollywood. He appeared in several films after he was discovered by Doris Day, but eventually got tired of movies and decided to move into the rapidly expanding medium of television. He turned out to have a knack for understanding the new formats that would work on television, and became a successful host of both talk shows and game shows. He once showed up in the audience at the musical at my high school (*High Button Shoes*, if I remember correctly) – apparently he had a niece in the cast.

Before long, he was not only a host of TV game shows; he was also a creator of them. One day, as he told the story, he was flying to New York from Duluth, Minnesota with his wife. He was trying to come up with ideas for a new game show, and his wife made a suggestion. In the 1950s there had been a series of scandals around the TV quiz shows. The most prominent such scandal involved the show *Twenty-One*, where it turned out that the outcomes were fixed beforehand. After those scandals, people had stopped creating quiz shows. Griffin's wife suggested that this might be a good time to propose a new one, perhaps with some kind of new twist to distinguish it. She thought she had an idea.

### **Jeopardy!**

Her idea was to turn things around: instead of providing a question and expecting an answer from the contestant, you provide the answer and the contestant has to provide the question. For example, if you provided the answer "1861", the contestant would have to provide the question "In what year was Italy unified?" To an answer "10 Downing Street," the correct question might be "Where does the Prime Minister of Great Britain live?" The contestant would always have to provide the response in the form of a question.

The idea was so good that it was accepted sight unseen by NBC, and the show was named *Jeopardy!* (the exclamation point is part of the name). It was launched in 1964 and became a hit immediately. The game was a straightforward quiz, with several categories of questions (or, I should say: answers) for the contestant to choose from, according to ascending monetary rewards. At the end of each show was a section called "Final Jeopardy," where each contestant had one last chance to influence the outcome by wagering up to his entire winnings on a single question.

That final question was accompanied by a catchy little tune that was timed to last exactly 30 seconds, the time that the contestants had to contemplate their final response. The title of that tune is "Think!" and it has an interesting story all its own. It was composed by none other than the versatile musician Merv Griffin himself – and was in fact originally a lullaby for his son. It has so insinuated itself into American popular culture now that it's used for just about any situation in which there is a countdown while waiting for something – for example, during discussions in baseball games or at the horse racing track, or even during Perry Mason or similar legal TV shows when awaiting a verdict. Merv Griffin liked to boast that for the 10 minutes he spent writing that little melody, he had received over 70 million dollars in royalties over his lifetime.

I once attended a Jeopardy show in the 1970s when I was a student at Yale. With a fellow student, I went down to New York and was part of the studio audience for a series of three shows and got to see the host Art Fleming in action firsthand. Back in those days, there weren't the fancy electronic displays there are today. Instead, it was literally a board with cardboard signs in the categories. When a selection was made by a contestant, a burly stage hand standing behind the board would simply pull off the piece of cardboard covering the question. For me, it was a bit like exposing the Wizard of Oz.

Later, when I was a student at Berkeley, I participated in an audition when the show's producers swung through San Francisco looking for new contestants. I didn't do so badly in the audition, but in the end they chose a sharp, good-looking young woman with a vivacious personality, and I can't really say I disagreed with their choice.

Jeopardy was an enormous international success, with adaptations in over 25 countries – including Italy in the early 1970s, with *Rischiatutto* hosted by Mike Bongiorno. Jeopardy wasn't the only successful TV game show that Griffin invented, by the way. The hugely successful *Wheel of Fortune* show was also his invention, and as we all know, it lived on with great success in Italy as the *Ruota della Fortuna*. He was eventually named by *Forbes* as the richest Hollywood performer in history, in large part due to his ventures in game shows.

So what was it that made Jeopardy so successful? It turned out that the clever inverse “answer/question” format encouraged extremely sophisticated uses of the English language, with many puns, shades of meaning, ambiguities, and subtle references that challenged the very best minds. The categories ranged over the entire spectrum of human knowledge, and so contestants had to be very well-read indeed. This was not a game for dummies – and it soon attracted a huge cult following. If that seems strange to you, consider the game show *L'Eredità* (“The Heredity”) that is running today on Italian television. Its final section, called “the guillotine” – in which the challenge is to guess a hidden word from among five words that are related in a way that the contestant must deduce – now numbers among its many fans no less a personality than the great Italian intellectual, professor of semiotics, and author (*The Name of the Rose*) Umberto Eco, who says he never misses it on TV (the show's producers once personally dedicated a “guillotine” puzzle to Eco).

In summary, a successful contestant on the Jeopardy show had to exhibit an extraordinary command of the English language in all its grammatical nuances as well as a mastery of a broad range of subjects ranging from history to geography to literature to current events and popular culture. It turned out that Jeopardy embodied exactly the type of challenges that were being confronted by IBM and others in the area known as *natural language processing*.

### **Natural Language Processing**

At this point, I'd like to go back once again to my old lecture and take up another topic and see how it turned out. Back then when I gave my lecture, nearly twenty years ago, there were essentially two opposing approaches being tried to making “intelligent machines”:

- Teaching things to them;
- Letting them learn for themselves.

The first approach involved the general idea of what were called *expert systems*. Suppose you wanted to make a computer that knew a lot about making medical diagnoses. The strategy was to find an expert in making medical diagnoses, and get all the knowledge out of his brain and into the computer – perhaps by interviewing him to

coax the expertise out of him. This approach made a lot of sense to everybody. After all, human beings are the source of the knowledge we're after, so why not go straight to the source?

Later on, this approach acquired even more credibility in the popular mind as the idea of the so-called Semantic Web emerged. Have you ever searched for a word or phrase on the Web and received the wrong answer because the meaning was ambiguous? For example, take the "Paris Hilton" problem: are you looking for a hotel in France or browsing for movie star gossip? Those who promoted the Semantic Web idea proposed that experts create various kinds of "dictionaries" inside the Web that can help sort out the different meanings a word or phrase can have. The more accomplished and capable the experts, the better the dictionaries will be, and the "smarter" the Web will become. This approach was adopted enthusiastically because it also seemed to lead directly and eventually (if not immediately) to the Holy Grail of computers that understand human language. And that made perfect sense, too: who was better qualified to teach computers to understand human language than humans themselves?

The second approach involved the general idea of "learning by example." Officially, this approach became known as Machine Learning, whereby an initially ignorant computer would essentially learn from experience. It was quite successful in some areas, especially controlling sophisticated machinery like robots or helicopters. Take the case of the helicopter: it's not that easy to get the control and balancing of a helicopter exactly right. So instead of trying to calculate exactly the right program to do the job, you just let the helicopter fly and see what happens. When the helicopter crashes, the control program considers that a bad outcome (thank goodness for that), and learns from the experience by avoiding the next time whatever it did to make it crash. (By the way, if you're getting worried: of course we're talking about unmanned model helicopters, not the real thing.) In time, after a certain number of crashes, the control program will have adjusted itself to do a far better job of keeping the helicopter in the air than a man-made program might have done. And in the control of robotic machinery the Machine Learning approach became almost universally the method of choice: you just let the machinery adjust itself after each "bad" outcome and it would eventually converge to the desired "good" outcome.

But a bunch of robots and helicopters learning how to move around and fly is a lot different from a computer learning to understand language. One is merely "motor learning," whereas the other is "cognitive learning." Why, they're not even on the same intellectual plane! Everybody knew that there was no way that Machine Learning, as effective as it might have been for controlling our manufacturing plants and vehicles, would ever lead to a way for computers to work with language, the highest expression of human intellect.

Indeed, in my lecture nearly twenty years ago, I dared to make the bold claim that in my opinion, finding a way to deal effectively with human language would be *equivalent* to solving the problem of Artificial Intelligence itself; because natural language, with all its complexities and subtleties, was the very embodiment of human intelligence. A computer could never "process" natural language if it did not truly understand it, I patiently explained to my audience that day in 1992.

And I was dead wrong.

### **The Rise of Machine Learning**

As the years passed, expert systems and the Semantic Web began to run into problems. Progress seemed frustratingly slow. It seemed to take forever to create those dictionaries of meanings by experts. One well-known dictionary of medical terminology (called SNOMED) ended up with over 370 thousand names at last count, and they're still not sure whether they have separate terms in there that really mean the same thing. Over time, many began to voice the opinion that the dream of the Semantic Web has not been realized and may never be realizable. And with that looming failure, the dream of being able to process human language seemed to fade.

But then something strange happened. Machine Learning began to have a string of successes in a most unexpected area: processing human language.

How could that be? What could have possibly happened to turn Machine Learning – that simple, primitive technique used mostly for teaching motor control to mechanical contraptions that walked or flew about – into a successful approach to dealing with language, that most sophisticated and subtle of all human characteristics that defines our very humanity? What had changed to make it all possible when it hadn't been possible before? What new and deep algorithms were discovered to capture the essence of language acquisition? Well, it turned out that it wasn't really the algorithms that enabled the breakthrough.

It was the data.

### **The explosion of online data**

One reason that Machine Learning had been so successful with teaching motor control to robots, helicopters, and the like was that they could have as many examples as they needed to learn from. Did the helicopter crash? Send it up again. Did the robot make the wrong move? Have it try again ... and again ... and again ... thousands of times if necessary, until it finally gets it right.

But human language is a different story. While it's true that humans acquire much of their language skill from examples, it's also true that they also have seemingly infinite resources for obtaining their examples – reading, talking, listening, studying, every day of their lives. Even if we thought Machine Learning of language through examples might be a feasible approach, where would all those examples come from?

In 1992, when I gave that lecture, the question was reasonable. The total amount of written language in electronic form was infinitesimal. Hardly anybody except software people like me used computers to write things. The Internet was small. Few people had e-mail. The number of websites was, literally, almost zero. The biggest stores of information in electronic form were databases full of numbers, not words – and even they weren't very large, all things considered.

But just a few years later, in the mid-nineties, things began to change quickly. With the arrival of the first web browsers like Mosaic, the number of websites began to shoot straight up. Today it is estimated that there may be somewhere around 150 million websites. And the size of the Internet itself? A recent study by IDC estimated that the Internet consists of around 1.2 *zettabytes*. That's 1.2 with twenty zeroes after it. It would take the entire population of the world sending SMS messages continuously for the next century to produce that much data.

Today we can find nearly anything in electronic form. The Wikipedia is by far the largest encyclopedia ever written, with over 3.5 million articles in the English language edition alone – and it's all right there online, available to anyone (and any computer) who wants it. There are books, journals, newspapers, dictionaries ... the list is seemingly endless. Suddenly humans don't have the monopoly on resources from which to draw examples for learning language. Computers have as many or even more – thousands, millions, even billions of examples drawn from the huge pool of information that is now in electronic form. But of course the real question is: does it make a difference?

It turns out that it *does* make a difference. Peter Norvig, the Chief Technical Officer at Google (and therefore someone who should know what he's talking about) has put it this way:

In the modern day, [with Machine Learning, we] observe how words and combinations of words are used, and from that build computer models of what the phrases mean. This approach is hopeless with a small amount of data, but somewhere in the range of millions or billions of examples, we pass a threshold, and the hopeless suddenly becomes effective, and computer models sometimes meet or exceed human performance.

It turned out that, in a way, Machine Learning of human language through examples was a bit like Einstein's Theory of Relativity. Relativity seemed impossible when it was introduced partly because nobody had ever had the experience of traveling at speeds near the velocity of light (except in the car of a certain Welsh friend). Similarly, the idea of a computer acquiring any kind of useful language proficiency through examples originally seemed impossible partly because nobody had ever had the experience of having millions or billions of examples to work with.

The irony is that in the end, Machine Learning turned out to work much more like the way in which we ourselves learn language than the expert-system approach does: in our everyday lives, we *also* observe how words and combinations of words are used, and from that deduce what the phrases mean. When I have just learned a new word in Italian, for instance, it doesn't mean much to me – it seems bereft of "semantics". It's just another arrangement of letters. But every time I hear it used by somebody or read it in a newspaper or a book, its meaning deepens for me, until eventually I can nearly feel its meaning viscerally as I speak it. I am literally learning through examples.

With the explosion in the amount of information in electronic form in the 1990s, Machine Learning started to replace the expert system approach to processing natural language. The learning process, or "training," was accomplished through large *corpora*. A *corpus* (that's the single of *corpora*) is a set of electronic documents full of examples

with the correct answers marked so that the machine can learn from them. A huge number of such corpora have become available in recent years. An example is the Open American National Corpus, which contains 15 million words of American English. It is all nicely annotated for parts of speech, verb stems, and the like, and is available to any person (or computer) for study. Another example is the British National Corpus, which contains 100 million words with samples of both written and spoken English from many sources. Yet another example is the Corpus of Contemporary American English (COCA), which contains over 410 million words. Using this ever-increasing pool of corpora for training examples, Machine Learning programs just got better and better and better.

As one success after another was obtained with this approach, companies began to realize the vast commercial potential behind having a computer that could process human language. One of those companies was IBM, which had created a so-called Semantic Analysis and Integration department to track developments in this area.

### **IBM Watson and the Jeopardy Challenge**

The destinies of the Jeopardy game show and IBM came together one evening in 2004 at a restaurant in which an IBM research manager, seeing an especially exciting round of the show being played on television, decided that this should become the next Grand Challenge. The next several years were spent in building a powerful computer for natural language processing based on Machine Learning, which they christened “Watson” – after IBM’s founder.

They fed Watson with every type of resource imaginable: literary works, dictionaries, news articles, databases, anything they could get their hands on – and they could get their hands on a lot, as I noted earlier. Naturally, they fed it the entire Wikipedia. By the time they were done, Watson had ingested 200 million pages of information.

It took a while to get off the ground. In its first tests in 2006, its performance was lousy. But just two years later it was holding its own against human Jeopardy contestants, and by 2010 Watson was ready for prime time. Arrangements had been made for Watson to compete on the Jeopardy show against the two most successful contestants of all time. Both IBM and Jeopardy executives realized the huge marketing value of this occasion and played it up with lots of publicity.

Then came the big day, less than two months ago [February 14, 2011]. As a computer filling an entire room, Watson couldn’t exactly stand up there at a podium next to the human contestants, so they gave him a snappy looking avatar that lit up as he “thought”. Since the game involved pressing a button when you thought you could respond correctly, he was also outfitted with a mechanical finger.

During the match, we were given an insight into how Watson works: during each answer/question session, the viewer would see the three topmost candidate answers that Watson was considering. For each answer, a meter showed the probability Watson was calculating that the answer might be the correct one. The one with the highest

probability and also passed a threshold was selected. Otherwise, Watson just shut up and let the others have their chance.

The point here is that Watson wasn't "thinking" in the same sense that we humans think: he was just calculating, like other computers do. That's how this kind of Machine Learning works: it's all based on estimating probabilities of things being right. In fact, it's generally called *statistical* Machine Learning for that reason. I'll get back to that later.

I'm sure you're all dying to find out what happened, so I won't keep you waiting: Watson won handily, winning over three times as much as each of the others.

### **The Aftermath**

So what happened after Watson's successful debut on Jeopardy? Certainly nobody claimed he had cheated, like Kasparov had done 14 years earlier with his predecessor. There were a few gripes here and there, like complaints about his speedy mechanical finger beating the others to the buzzer, but overall there was general acknowledgement that he had won, fair and square.

But what everybody *did* seem to wonder was, "What does this all mean?" There was article after article in nationally syndicated newspapers, and interviews with analysts on TV talk shows, discussing whether this all meant that intelligent machines were about to take over the world. A month ago [28 February 2011] the *New York Times* carried an invited article on this subject by none other than Hubert Dreyfus, the philosopher from the University of California at Berkeley whom I had discussed at length in my lecture nearly twenty years earlier on the same subject. Here is what he had to say back then:

Great artists have always sensed the truth, stubbornly denied by both philosophers and technologists, that the basis of human intelligence cannot be isolated and explicitly understood.

His essential argument involved the fact that humans have bodies and therefore a context in the world around us, whereas computers don't. Here is an excerpt from what he had to say in the *New York Times* article, where he was analyzing the reason that Watson failed to understand the relevance of a subtle clue during one of the sessions:

... Watson doesn't understand relevance at all. It only measures statistical frequencies. The fact is, things are relevant for human beings because at root we are beings for whom things matter. Relevance and mattering are two sides of the same coin. As [the philosopher] Haugeland [has] said, "The problem with computers is that they just don't give a damn."

That didn't impress one of the readers of the article, who commented that

... many of our politicians, criminals and military people [also] lack understanding of human beings and seemingly have no compassion for humans.

In summary, nearly all of the philosophers, analysts, and commentators agreed that Watson didn't represent a revolution in Artificial Intelligence. Yet they also universally agreed that it had accomplished something very impressive. I can think of no better way

to put this into the proper perspective than to go back once again to something I discussed in my lecture of nearly twenty years ago.

Artificial intelligence has sometimes been defined as the use of computers to solve problems that previously could only be solved by applying human intelligence. Now, the problem is that this definition has a sliding meaning, as the computer scientist David Parnas has noted. In the Middle Ages, it was thought that arithmetic required intelligence. Now we realize that it is simply a mechanical process, and we've built pocket calculators to do it for us.

In the 1990s chess was only the most recent of those tasks we thought were in the exclusive domain of the human mind. Likewise, we had never thought that a computer could do anything useful with language by mere mechanical computation – surely *that* was an exclusively human activity. And yet, Watson (and many other systems like it) had shown that it was possible. Watson had literally and figuratively beat humans at their own game.

### **What's Next?**

So just what *are* the possibilities opened up by the conquest of natural language by computers? We can start the discussion with Watson himself, which is a so-called *question-answering system*: you ask it a question and it gives you an answer (except in Jeopardy, of course, where you ask it an answer and it gives you a question).

I presented one of the very first question-answering programs during my lecture back in 1992. That program was called Eliza (after the character in *My Fair Lady*), and imitated a psychoanalyst. Lucia Ghiselli and I together read the script of one of Eliza's most famous conversations (Eliza eventually decided that Lucia had a problem with her boyfriend; she probably got into trouble at home that evening).

But there are many, many much more serious applications for question-answering systems, and that is one reason IBM created a Grand Challenge for itself in that area. After the Jeopardy match, IBM began a campaign to publicize the great future Watson was going to have. And the first application they put up in lights was the very application we talked about earlier: an expert medical diagnosis system (those poor expert systems people must be gnashing their teeth at seeing those primitive Machine Learning people succeeding in just the area they were working on much earlier).

Here's how Jennifer Chu-Carroll, an IBM researcher on the Watson project, put it in an interview with *Computerworld* magazine:

Think of some version of Watson being a physician's assistant. In its spare time, Watson can read all the latest medical journals and get updated. Then it can go with the doctor into exam rooms and listen in as patients tell doctors about their symptoms. It can start coming up with hypotheses about what ails the patient.

(I wonder what else Watson does "in its spare time" ... dream of electric sheep?)

A well-read Watson-like system could answer questions and make recommendations in a number of fields. You may not be surprised to hear that the financial industry is

looking into the idea of using Watson-like programs to get good stock tips. And of course the military is *always* interested for lots of applications they prefer not to talk about (and maybe we'd prefer not to know about). A company right here in Pisa has created a question-answer system that is being used by the Italian telephone companies to answer questions by customers about their services, saving a lot of time for harried call center operators.

### **Coping with the Vastness**

Question-answering isn't the only opportunity for computer language processing. The possibilities are as vast as the Internet itself. In fact, in large measure they are vast *because* of the vastness of the Internet.

I mentioned earlier that the size of the Internet today is estimated at about 1.2 zettabytes. But incredible as that number seems, it's just the beginning: by 2020 the size of the Internet is predicted to be over 35 zettabytes. But most importantly (and relevant to this talk), over 70% of that data is being generated by individuals like you and me, with our social networks, blogs, tweets, and e-mails. And most of that is plain old text written in English, Chinese, Italian, and all of the other human languages around the world.

To put it simply, the Internet has gotten so large that it is beginning to be beyond our capabilities to manage it. Much of this is our fault, of course: we regularly receive messages (especially jokes) that we forward to all our friends, duplicating them many times over; we chat endlessly on the social networks and forums; one of my brothers once proudly told me that he had never deleted a single e-mail message. The result of all this is that nobody can read – much less find – everything that he needs to any more. We need help. And computers that can process language can provide that help.

With the advent of natural language processing, a new discipline known as **text analytics** has emerged. Text analytics help us sort through all the masses of documents, messages, words, phrases – everything that's written down out there in the Internet – to help us do whatever we're trying to do. I'd like to present a couple of examples now of what we might be trying to do.

### **What are they saying about me?**

Those of you who have heard my last four or five lectures will remember that they have always somehow, some way, involved the *social networks*. That is the measure of how central they have become in our lives. I know people (and I'll bet you do, too) who track their entire day on Facebook or Twitter or some forum, discussing everything you could possibly imagine: what they had for breakfast; the beautiful new Gucci shoes they just bought; how their sleeping pills made them groggy; what a great job the President is doing; what a terrible job the President is doing. You name it, and you can be sure that it's being discussed out there on the social networks.

And that makes the social networks a powerful force in every aspect of life now – *including* business life. Consider the case of Canadian musician David Carroll and his

band, the Sons of Maxwell. At a stopover in 2008 at the Chicago airport on a United Airlines flight from Canada down to Nebraska, he looked out the window to see the baggage handlers tossing guitars – one of which was his – onto the tarmac like they were sacks of potatoes. Sure enough, his 3500 dollar Taylor guitar turned out to have a broken neck when he arrived in Nebraska. (Now, there is at least one professional guitarist in our association, and several of us are amateur guitarists, so we can all appreciate this. In fact, many years ago on a flight between New York and San Francisco, my own guitar arrived with a hole in the back, punctured by something that went right through the case. I played it again just a few weeks ago [early March 2011] – that hole is still there.)

You can imagine the wall of indifference Carroll faced when he tried to complain to United about it. So he decided to take matters into his own hands. He made a music video about his experience, composing a clever little tune and lyrics just for the occasion, entitled “United Breaks Guitars.” He uploaded the video to YouTube and sat back to see what happened.

What happened was that it was viewed by 150 thousand people in *one single day*. Three days later that had risen to half a million. Just over a month later it had been seen by 5 million people. In short: the video had gone viral.

As you can imagine, it was a public relations disaster for United Airlines, and they rushed to make amends as quickly as possible. As for Carroll, he got a great boost from the incident, too, and became a sought-after speaker on customer service. (On one of those speaking trips in 2009, United managed to lose his luggage. One does wonder ...)

The moral of the story is that, for businesses, much of what is known as “customer relations” has migrated onto the social networks and you had better be aware of what people are saying about you out there. But there’s just too much of it. Nobody can sit down and read everything they’re saying about you, much less even find it. That is where something called **sentiment analysis** comes in.

Nowadays it seems that everybody has an opinion and is happy to express it in a very public way. As soon as people do *anything*, they get out there on their favorite social network and write a review. You can see reader reviews of books on Amazon.com; traveler reviews of hotels and restaurants on TripAdvisor.com; reviews of products of all kinds on specialized forums all over the Internet. Sentiment analysis systems read texts like these reviews, or complaints, or whatever might be fed to them, and they try to determine whether something positive or negative is being said. As you can imagine, they won’t always get it right, because language can sometimes be convoluted and idiomatic (when Michael Jackson called something “bad” he was probably telling you that it was good). But these systems can still read a lot more than you can, and get a sense of whether things are going well or not. If you’re anybody whose reputation is a key part of his business (such as the fashion industry), then you have to get out there and protect your brand name in the blogosphere.

## **e-Discovery**

Protecting one's reputation for making nice clothes and serving good food isn't the only motivation people might have for sifting through lots of electronic information; another pretty good motivation is staying out of jail. In lawsuits, there is a task known as "discovery," where lawyers look for documents that are considered relevant to the case. Now, this was already a rather boring and frustrating job back in the old days when most documents were on paper. But now, when most documents are in electronic format, in every conceivable form from text files to spreadsheets to e-mails, it's often downright impossible. There's just too much out there. So-called "e-discovery" systems are out there now, reading the documents and deciding whether they're relevant to a case. And those programs don't get tired and bored like we do.

It's not just lawyers who have to worry. Pharmaceutical companies have a legal obligation to be aware of complaints of people who have used their products and had problems (like, say, a sleeping pill making you unusually groggy). If somebody has made that complaint publicly on a social network, then it is definitely in your best interests to be aware of it or you will find yourself with a lot of explaining to do after an incident occurs. But here, too, the job has become simply too big for people to do by themselves.

Text analytics systems are able not only to sift through masses of electronic documentation, but they're also able now to do some pretty clever sleuthing with what they find. They can read through piles of seemingly unrelated newspaper articles and piece together indirect or even intentionally hidden relationships among persons of interest (for example, two people might be trying to hush up the fact that one of them has a financial interest in the other's company.) They can look for suspicious patterns that might indicate there is some funny business going on (for example, if somebody closes a bank account in one country and opens another one in a different country on the same day, it may indicate some financial hocus-pocus that deserves a closer look by the authorities).

This is all very impressive, but it's also rather worrisome from one particular point of view: human employment. The tasks I just mentioned have been performed up to now by highly qualified – and well paid – humans. As much as we had gotten used to the idea of computers automating menial tasks like adding numbers and assembling automobiles, nobody ever expected them to encroach into this area. Here's what Tom Mitchell, chairman of the Machine Learning department at Carnegie-Mellon, had to say about it in the *New York Times* a few weeks ago [4 March 2011]:

The economic impact will be huge. We're at the beginning of a 10-year period where we're going to transition from computers that can't understand language to a point where computers can understand quite a bit about language.

Who would have ever thought it ...

## Conclusion

I certainly wouldn't have thought it back then in 1992 when I boldly stated that mastering computer understanding of human language would be equivalent to solving the entire problem of artificial intelligence. But my mistake wasn't in making that bold statement – in fact, most researchers still think that statement is true. (The official way to express it is to say that computer language understanding is “AI-Complete”). No, my mistake was in thinking that the problem of computer language understanding had to be completely solved before anything useful could be done. It turns out that a *lot* of useful things can be done by computers with language without having to fully understand its meaning.

Sure, that means that the Holy Grail of perfect translation of idioms, puns, and poetry is still beyond the reach of computers – that task requires full understanding and may never be fully realized. But that's not the right perspective on what is happening here. Rather, it reflects the longstanding tension between two different approaches to the relationship between human beings and computers: “AI versus IA.” The first, Artificial Intelligence, was championed by pioneers such as John McCarthy, and focused on building *machines that think*. The second, Intelligence Augmentation, was championed by other pioneers such as Douglas Engelbart (who, among much else, gave us the computer mouse), who focused on building *machines that help people think*. In the “IA” approach, it doesn't matter whether Watson really understands what he's doing; all that matters is whether he's doing something that is useful to us. In this approach, computers will always be our assistants, not our masters. Of course, it's unnerving to see how much our assistants can do now – but they remain our assistants nonetheless.

## Epilogue

After writing those last words, I decided to reassure myself by going back to the phrase that provided the title for this talk, and seeing how well a computer translation program would do today. Nowadays that's easy to arrange, because some of the very best programs are right there online and freely available for all to try out. Google Translate is an example of a modern translation system that uses the same “statistical machine learning” techniques that I have presented and discussed during this talk. So I fired it up, set the “input language” to English and the “output language” to Italian, and I fed it the phrase:

The spirit is willing but the flesh is weak.

In return I received:

Lo spirito è pronto ma la carne è debole.

Oh my ... that's not bad at all ...

## **Resources**

YouTube: you can find all the television sessions of the IBM Watson Jeopardy Challenge online at YouTube. You can also find an entire course on Machine Learning courtesy of Stanford University, although it is not for the faint of heart.

Stephen Baker, *Final Jeopardy*. The author followed the IBM Watson team around as it prepared for its Jeopardy Challenge, then wrote a book about it.

IDC, *2010 Digital Universe Study*. Estimates and ruminations on the size of the Internet, now and future.

Stephen Marsland, *Machine Learning: An Algorithmic Perspective*. A textbook for those who want to know more about machine learning.