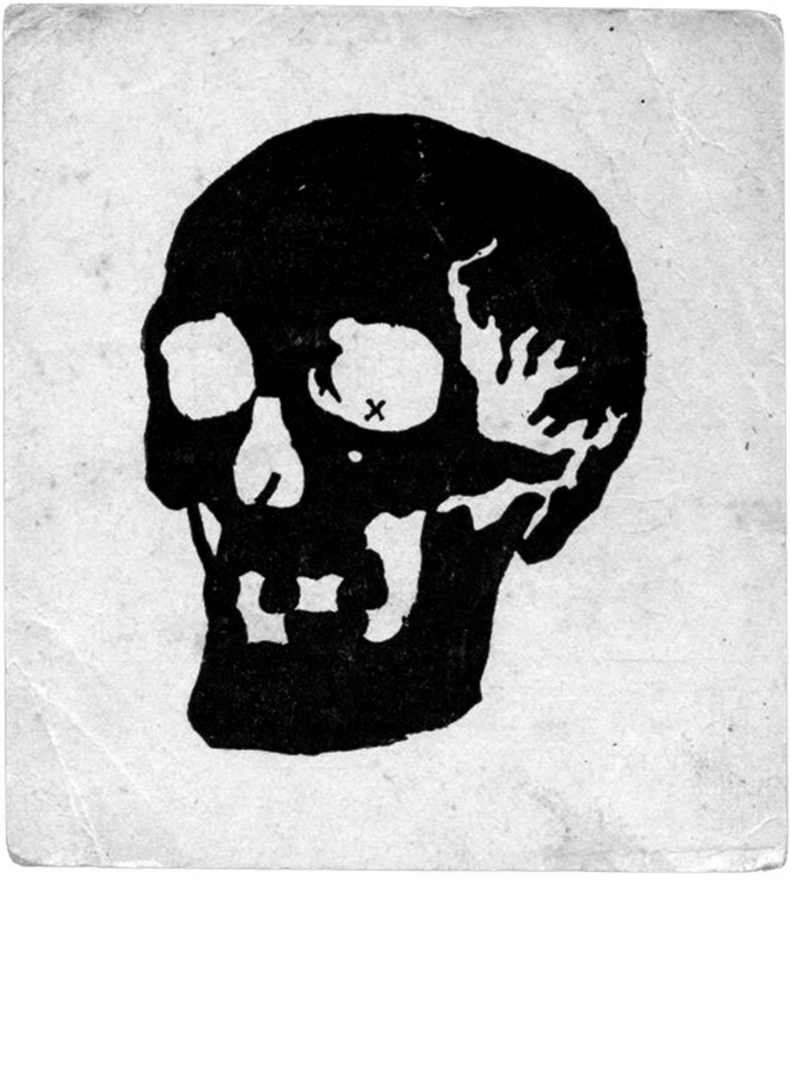


This bulletin was conceived, sort of, on the road from Copenhagen to the town of Helsingør and Kronborg castle, the non-fictional setting of William Shakespeare's fictional play *Hamlet*. With thanks to the Royal Library of Copenhagen, whose circulating collection of books by Charles Babbage are first printings, inscribed and annotated by Babbage himself.

The skull on the cover of this bulletin and the directions for seeing Yorick's ghost in your mind's eye (on the back cover) are reproduced from two sides of a Victorian trading card manufactured by British soap maker, Pears.

The image on p.124 was posted to Wikimedia Commons on August 18, 2010, by Alan Levine of "Strawberry, United States." It appears here under a Creative Commons attribution license.

BoTSL#3 2012 APR 15 4:49 PM

Here's a Soviet-era joke told by one computational linguist to another:

Have you heard that Academician Ambartsumian has just won a Volga car in the state lottery?

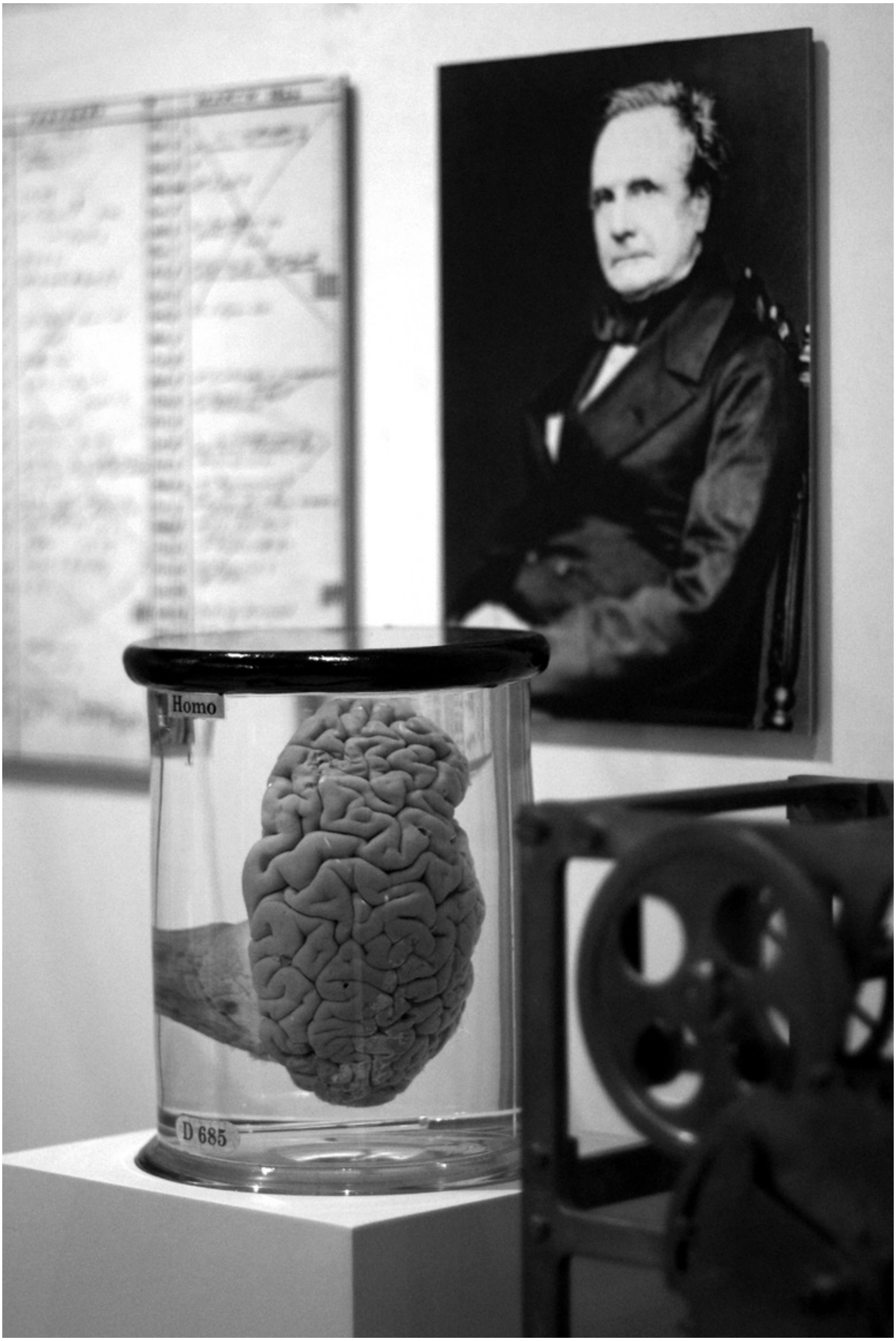
Of course I have. Only he's no academician. He's a night watchman. And his name is not Ambartsumian. It's Rabinovich. And it was not a car. It was a hundred rubles. And he played poker, not the state lottery. Oh, and by the way: he didn't win!

It's borderline uncivilized to explain a joke, but this is an essay, not a party, so bear with me. First, you'll need to know what computational linguistics is. In the slimmest of terms, it's a subfield of computer science and linguistics in which theories about language are implemented as procedural operations. That means it's a research area for generating and testing hypotheses about how language works. The testing is done by Artificial Intelligence (AI) applications intended to process language in ways that seem, well, human. The field involves theory as much as practice—in the overlap of our natural language and the language of formal logic on which our computing machines are based, it's not altogether clear which language envelops the other, hence the need for theoretical speculation. And therein lies the joke. The Russian who responds, "Of course …" treats the other's question as a formal structure, within which specific words are interchangeable variables. He acts as if the sense of the question is foremost its logical form, not the difference between, say, academics and night watchmen or toaster ovens and microwaves. He disregards the coherence between structural and representational aspects of language.

Here's a visual analog of what the joke indicates by exclusion:

Neither the black cross nor the white cross exists except that the pinwheel exists except that the pinwheel exists only because both the black cross and the white cross exist. We can see the white cross and the black cross and the pinwheel at once, while no one aspect of the image could be seen without creating the other two.

BoTSL#3 2012 APR 15 4:49 PM


Homo
D 685

The pinwheel is an ambiguous figure borrowed from Ludwig Wittgenstein's *Philosophical Investigations*. The joke is borrowed from "What's in a Symbol: ontology, representation and language," a scientific paper written in the form of a dialog. Its authors contrast seemingly irreconcilable points of view regarding symbolic representation in AI—specifically, that words in a formal language are, in fact, *words,* or that they are *symbols,* like the natural language words they exactly resemble. If a word is a word, then the abstract nature of a word as a symbol for something else is in doubt, and vice versa.

All this may stink of mumbo jumbo if you don't earn your keep telling computers how to do their jobs and you haven't opened a grammar textbook since eighth grade. But chances are, if you happen to talk and use computers, and certainly if you happen to use computers by talking to them, you've engaged with the work of computational linguists. Their theories about the intersection of mind and machine only become more relevant as we approach near-total integration of computers in daily life. By now, one could even envision a future history of computing that would tell how we one day came to understand language mathematically. Or not. It isn't necessarily possible. To grasp why not, and why it matters, some backstory is in order.

. . .

Inside a jar inside a vitrine inside the computing gallery at the London Science Museum, the right half of Charles Babbage's brain floats languorously. At some point, it migrated to its current station from the Royal College of Surgeons, who received it upon Babbage's death in 1871. I learned what I know about the specimen from a nearby title card, which explains that dissection of revered men's brains was common practice at the time. While it may be evident today that less about human intelligence can be gleaned by cutting into a decommissioned brain than by monitoring a live one (and little, if anything, is discovered about intelligence even then—whatever intelligence is), one can understand why the College did what they did to Babbage's left half. The work of hearts, guts, lungs and joints compares to that of pumps and hinges. Why shouldn't minds have their mechanistic parallel, too?

Babbage was born in London around the time of the French Revolution. He died 80 years later, a mathematician, professor, philosopher, inventor, and gadfly. Today, he's best remembered as the father of the computer, though a German engineer, Johann Müller, is thought to have been the first person to come up with the idea. In a postscript to a letter written a few years before Babbage's birth, Müller described a clockwork-inspired machine that would calculate numerical tables and print the results without human intervention. A good logarithmic table was a precious item. It could be used to figure out how to get from one point in time and space to another, or to increase accuracy and speed in further calculations, like those required for land surveys and taxation. The navigational and bureaucratic agility afforded by a high-quality table yielded economic and political advantages, but precise tables took years to prepare, check, and duplicate by hand, then typeset and print. At each stage, humans were prone to err, not only in computation, but in the laborious acts of copying and publishing that followed. The mandate of the mechanical computer, then, was not to *think* like a human, it was to be more reliable than one at crunching numbers and making copies.

Babbage called his machine a Difference Engine. When I picture what that could be, I see a combustion-powered behemoth churning out miscellany: one minute a treatise; the next minute a donut; never the same thing twice. For Babbage, a "difference" wasn't some nebulous conceptual goo separating discrete ideas, it was simply a sum, as in the quantitative difference between two numbers. His engine was an automated means for calculating and recording sums that would limit human involvement to initiating a query mechanically and turning a crank. Babbage conceived of the thing in 1822, then failed slowly, expensively and cantankerously to build one over the following decades. He had many talents. Engineering and diplomacy ranked low among them. A Swedish father and son team completed several versions of the engine between 1843 and 1859, but these were a commercial flop (as Michael Lindgren documents in *Glory and Failure*, his gripping history of numerical tables).

Babbage's brain had been shelved for well over a century when construction of his revised design, Difference Engine No. 2, began in the 1990s. The first functioning No. 2 was completed in 2002, another in 2008. They cost millions to produce—a price that registers historical intrigue rather

BoTSL#3 2012 APR 15 4:49 PM

than computational prowess, since demand for dumpster-sized mechanical calculators fell from almost nil to pretty much nil by the mid-1970s, when handheld electronic ones began to catch on. One of the engines now resides in the Science Museum, just a few steps from Babbage's right lobe, alongside other relics from the nascent computer age. The second is housed in the Computer History Museum in Mountain View, California, where it is put to use for public demonstration by a couple of retired engineers several times a week.

I attended a performance of the California engine last year. Ignition was delayed a few hours while a technician repaired No. 2's printing apparatus, so I spent a full day at the museum, watching every film, reading every wall card, following the docent-led tour, loitering in the bookshop, and finally camping out in a small chess exhibit near Babbage's gimpy apparatus.

The fabled story of computer chess picked up steam in the 1950s and didn't let off until 40 years later, when IBM's Deep Blue (née Deep Thought), defeated world champion player Gary Kasparov in 1997. One of the first chess programs was designed to economize computation by a combination of procedures for testing possible moves (algorithms), honed according to generally accepted rules of thumb (heuristics—a knowledge base). In theory, the program's success would depend on how well it assessed best possible moves for itself and worst possible moves by its opponent in the shortest amount of time. In theory, effective pruning of the search would eliminate less likely possibilities relatively quickly, increasing the probability of finding a strategic move in an allotted turn. In fact, the program's success was greatly exaggerated. It played its last recorded game in 1960, two years after it was first wheeled out; a novice ten-year-old child vanquished it in 35 moves. But hype kept air in the balloon of burgeoning research, and the program's model proved influential.

The first official match between computer and human in a tournament occurred two years before the Apollo moon landing, in 1967. Over the next three decades, engineers from rival labs developed programs to compete at annual events, using ever-improved hardware capable of performing an ever-greater number of operations in ever-shorter amounts of time. Competitions pitted automated players with names like BELLE and CHAOS against each other in live matches before public audiences. Belle was developed in 1977, at Bell Labs, using a "brute force" approach to decision-making. Think of brute force as performing as many tests as necessary with a simple algorithm, versus the heuristic method of first reducing the test pool with the aid of stored knowledge. Brute force is simple but inefficient. The advantage is that it will find the correct result, given enough time. So its viability increases with processing speed and decreases with larger test sets. Belle's inventors designed special hardware to streamline its processing, thereby increasing its speed, and Belle, at a cost of under $20 thousand, won in a match against a $10 million supercomputer in 1982.

The story of computational linguistics began around the same time as computer chess, when many people assumed language would be a more tractable problem. Think: Most people speak. Relatively few are good at chess. Chess must be harder. And for humans, it is, generally. The diabolical computer featured in Stanley Kubrick's 1968 film, *2001: A Space Odyssey* (based on a story by Arthur C. Clarke), was a master of chess *and* conversation, but as the film nears its 50th birthday, even the most advanced computer dialog capabilities lag well behind HAL's. (Incidentally, HAL is an acronym for Heuristically-programmed ALgorithmic computer.)

. . .

Warren Weaver was a mathematician and engineer keenly interested in probability and statistics research. In 1949, he co-authored the book version of Claude Shannon's *The Mathematical Theory of Communication*. During that same year, he issued a memorandum to 30 colleagues, entitled simply "Translation," its contents seeded in an earlier letter to cyberneticist Norbert Wiener. Essentially, the memo called for application of "electronic computers" to the problem of international language translation. Weaver reflected on wartime cryptography research to hazard that certain aspects of languages might be universal, given that encrypted messages could be decoded by cryptologists who were not necessarily speakers of the encoded languages. In his words, "one could hardly avoid guessing that this process made use of frequencies of letters, letter combinations, intervals between letters and letter combinations, letter patterns, etc., which are to some significant degree independent of the language used." Weaver was well connected and well respected. Government interest and funding answered his call.

Strong swimmers waded forth, but soon found themselves head under in dark water. Early efforts in Machine Translation (MT) focused on word-for-word matching, primarily between Russian and English. It was immediately obvious that dictionaries and thesauri must be made machine-readable, but definitions are ambiguous. Take a word like "field." Could be a noun. Could be a verb. It has several unrelated definitions in either case. To identify specific usage in a particular phrase or sentence, one would need to know the part of speech and the relationship of the word to other words around it, each of which must also be identified in a similar way. Not only that, dictionaries and thesauri are themselves created by humans, and their overarching organization embeds certain patterns reflective of their makers. This information, too, is consequential. Progress in translation would require more precise lexical and syntactic characterizations; as well as formal characterizations of logic and grammar; algorithms to apply these descriptions; and hardware capable of storing the knowledge and processing the algorithms. That's just for starters—much more has been discovered about the difficulties posed by linguistic analysis since the first attempts to automate it.

U.S. funding for MT increased in the late 1950s, after Sputnik, but the work proceeded slowly and initial successes didn't lend to hype. In 1966, an infamous report by the Automatic Language Processing Advisory Committee recommended against further government funding of MT on the grounds that research showed little promise of producing quality translations of Russian documents quickly and cheaply and soon. As a result, funding was cut, and interest in MT declined over the next ten years or more, particularly in the U.S.

Today, the software component of computational linguistics risen from MT's ashes persists as Natural Language Processing (NLP). Whereas the goal of MT was to translate between human languages, the goal of NLP is to develop translation faculties between humans and machines, an ambition that pertains to philosophy, psychology, neurology, and cognitive science as much as linguistics and computer programming. It touches

BoTSL#3 2012 APR 15 4:49 PM

almost everything we already do on the Internet along the lines of tagging, searching and other means of annotation and retrieval, as well as higher-order tasks like speech recognition and international language translation—which are still largely unresolved. NLP combines other processes in less familiar applications, like avatar-led computer-to-human dialog, replete with simulated affect, designed to gather information from a human-user through spoken exchange and build a knowledge base to serve ongoing conversation and information gathering, in turn. I'm talking about computer programs designed to befriend you, if you can imagine being friends with a programmatically sympathetic listener who records and analyzes everything you say. Their inventors call them "intelligent and sociable companions." Benevolent prospects include keeping lonely people contented. More disconcerting prospects include imitating dead people.

Effective natural language simulation in live conversation with humans is the holy grail for NLP, but the task is not yet well understood. Computers were designed to outdo us in math. They were designed to eliminate the capacity for error in computation. They were designed to replace human beings in their role as computers—that is, as people who perform computation. Everything a computer does is based on the unambiguous difference between true and false, on and off, 1 and 0—in a word, a bit. Computers combine bits according to algorithmic instructions. This sounds so basic as to be incoherent to those of us who operate computers at a level far removed from the material process of binary computation. Notwithstanding the relatively small population of programmers concerned with atoms of machine language, most of us interact with computers through words and word-like symbols. Increasingly, we even speak to computers in our language. But there is a difference in magnitude between programming a computer to record and transcribe dictation or to execute a "save" (which is just a command to write certain bits to computer memory), and, say, programming a computer to parse a sentence for meaning then respond intelligibly in natural language. The problem may come down to a qualitative difference between words and numbers, somewhat more like the gulf between treatise and donut than sum and sum.

This quibbling is confusing, especially for seasoned researchers. Much NLP work is predicated on the assumption that words in an AI programming language are very different from words in a natural, human language.

Think of the first case as word-words, more like algebraic variables specifically arranged in a predictable structure, and the second as word-symbols, more like representations of things in the world, which occur and mix in a haphazard way, though even these definitions are meekly provisional, and many NLP researchers wouldn't allow them. The distinction is slightly easier to draw out by considering the implications of either case for an AI model of semantic parsing. Of course, "slightly easier" is almost meaningless here since the alleged ease of explication compares to squeezing some ineffable quality into a sayable nutshell. Further, whenever we talk about models of semantic parsing, it should be acknowledged that the supposed human method is inconveniently unknown, and perhaps shouldn't even be considered in the singular, since there may be as many human models for processing meaning as there are humans, all of them buzzing in flux around some ever-shifting middle ground of understanding, only if, only when, and only just as long as absolutely necessary. See? It's confusing already, and I've barely cracked the lid. Back to AI: if we assume a word is a word (not a symbol for something else), semantic processing could be a matter of manipulating a knowledge base of arbitrary signs (word-words) in a structurally predictable (algorithmic) way. This model may be good enough to get computers close enough to processing natural language that it wouldn't matter whether or not they would be capable of constructing natural language on the fly, as humans are. For most day-to-day applications, the difference may not make a difference.

But consider it again from a slightly different squint on the technical spec. When we read, the words we see are haunted by ghosts of what we already know. To demonstrate: computers were created to replace computers. The difference you read between the two instances of the word "computers" is informed by the contextual relationship of the sentence to the preceding sentences and paragraphs. That is, a meaningful difference between two consecutive instances of the same word has been constructed by me in the course of writing these pages. Seems simple until you try to imagine what might be involved in creating a practicable set of instructions to codify and anticipate just this one effect, and not only anticipate it, but produce it. Words (word-symbols), unlike bits, are inherently imprecise information. Computers can't sleuth and emulate certain linguistic mysteries yet because humans haven't successfully described how those mysteries unravel in formalist terms—the only terms a computer can manipulate. Whether or not complete formalization is even possible remains a matter of conjecture.

Charles Babbage described the difficulty ghosts pose to logic in 1826, in his presciently titled book, *On the Influence of Signs in Mathematical Reasoning*:

In the language of analysis, it is very rare that any symbol possesses more than one meaning; in ordinary language, it is as rare to find a word having but one signification ... **in common language, the meanings of words shade away into each other, and it is frequently difficult, even on mature consideration, to assign the precise limits of the signification of words which are nearly synonymous.**

A government official once asked Babbage whether his Difference Engine would give the correct answer if asked the wrong question. Babbage, not one to suffer fools lightly, couldn't fathom this muddle of misapprehension. Yet, thinking back to Kubrick's *2001*, or forward if we clock from Babbage's day, it's easy for us to imagine an absent-minded Dave wandering into the control room wearing the mildly bewildered look of someone whose short-term memory has just lapsed, blurting, "What did I come in here for, HAL?" to which HAL might reasonably reply, "Your glasses are on top of your head, Dave." Granted, there's an elbow missing between the Difference Engine shoulder and the HAL wrist, which fiction conveniently ignores. We still don't know exactly how the joint should be put together. All the while, NLP researchers go on moulding makeshift prosthetics.

Filling in the blank would be a matter of rigging a strictly logical processor with no capacity for interpretation to simulate interpretation. Or of persuading a machine with no capacity for belief to believe in ghosts. (That is, to recognize the second formulation as an anaphor to the first, an anaphor which refers to a machine metaphorically as a sentient being—one susceptible to persuasion—and extends another metaphor from the preceding paragraphs, which equates the formally irreducible ambiguity of words with a supernatural presence, etc., etc., etc.) Naturally, one might think that thinking of words as word-words necessarily entails a strict formalist approach to NLP, but even this isn't necessarily so.

BoTSL#3 2012 APR 15 4:49 PM

. . .

Recall Ambartsumian and Rabinovich, the academic and the night watchman from "What's in a Symbol" (published in 1999). One of the authors of the paper is a leading researcher in the area of artificial social companions. His name is Yorick Wilks. You may know Yorick as the man of infinite jest remembered fondly by Hamlet in Act V, the graveyard scene. A lonely prince addresses Yorick's empty skull, utters grief. It can be no coincidence that someone named Yorick is behind artificial companionship, unless a word is only a word. (The other author, Sergei Nirenburg, plays accordion but does not specialize in pseudosympathetic companions.) Wilks completed his doctoral research at Cambridge in 1968, under philosopher R.B. Braithwaite and linguist Margaret Masterman. Masterman had been a student of Wittgenstein and was, herself, a pioneer in MT, leading the early work on thesauri. Wilks' thesis entailed machine analysis of metaphysical arguments from major philosophical works. Looking back on his dissertation in 2006, in a paper called "Ontotherapy, or how to stop worrying about what there is," he wrote that his claim

was that such argument proceeds and succeeds by methods quite different from the explicit, surface, logical argument structure proposed by its original philosophical authors, but rather by methods involving rhetorical shifts of the senses of key words and of which the author may not even be aware e.g. the whole rhetorical purpose of the philosophy of Spinoza, set out in the form of logical proofs, all of which are faulty, is actually to shift our sense for the word "nature"...

On the basis of Wilks' analysis, one might suppose Spinoza's philosophy boils down to rhetorical sleight, but that isn't what Wilks supposed. His conclusions raise questions about how language works, which is different from raising questions about how language SHOULD work. He continues,

It was this early investigation that alerted me to the possibility that representational structures are not always necessary where they were deployed, and that it is hard to be sure when representations are or are not adequately complex to express some important and complex knowledge, and that one should be very wary of the usefulness of logic-based formalisms where language is concerned.

What is the use of rhetoric if humans are persuaded exclusively by logical argument? And what of proofs if logically unsound rhetoric will do? Must Spinoza's proofs be sound in order to shift our sense for the word "nature"? Wilks' pinwheel of conclusions contradicts a major tenet of the larger AI research community: that logic-based knowledge representation is universally powerful. Change "is" to "should be" in that last clause, and it's easy to understand why intelligent machines like HAL populate dystopian science fiction. By design, to be a wholly logical operator is not to be human. Only lately have we desired computers to be more like humans than pocket calculators, but how (and whether) to go about adapting a Difference Engine into a personable, talking companion is not obvious. The impetus behind the invention of the computer was the desire for a wholly logical, inhuman operator, not an automated friend.

Wilks has been involved in computational linguistics almost as long as the field has been in existence. His current work utilizes computers to capture and transcribe our speech, to detect and mirror our emotional register, to talk to us, to ask us questions, to extract our memories, to write our stories, to comfort and advise us while we are alive, and, potentially, to simulate some aspects of us after we die. Dialog capabilities are still rudimentary, but to the extent that a computer can be programmed to accept natural language as input, process it, and render it as output, language can become an interface, which means the physical components of computer interfaces as we now know them could take on other forms, or practically disappear. If a machine could one day accompany us over the course of our lives, simulate interest in us so as to earn our trust and emotional dependence, then cull and analyze our expressions, in the end—the end of us—its perfect memory of everything we represented to it would be more like our brain in a jar than our brain in a jar.

Automated social companions evolved from a motley bunch of misfits. An early dialog program called PARRY (1972/1973) simulated a paranoid patient in a Veteran's hospital. Parry had fear and anger metrics. He would react badly when these were high. He was prone to rant about the mafia. Unlike his better-known predecessor ELIZA (1966), who simulated a psychotherapist by repeating or mirroring a user's inputs, Parry did not parrot. He operated according to a process of pattern matching. Against a table of 6,000 patterns, he could analyze a user's inputs approximately and respond with greater specificity than Eliza, who drew from a collection of stock phrases to keep her conversations moving. Eliza can be found online. Here’s a transcript of our session:

> Hello, I am Eliza.
* Hi
> How are you today. What would you like to discuss?
* My feelings
> Tell me more …
* I’m sad
> How long have you been sad?
* About half an hour
> What does that suggest to you?
* I need a cookie
> Say, do you have any psychological problems?

Parry is not available online, but he met Eliza in 1972, and their conversation was recorded:

> What else would you like to discuss?
* I am getting tired of answering the same questions over and over […]
> I am not sure I understand you fully.
* You harp on the same questions too much.
> What does that similarity suggest to you?
* I don’t get you.

Many relevant problems in NLP have been apparent since the first decades. Eliza and Parry demonstrate that competing models obtain within the field, as was true for their chess-playing counterparts. Looking back in 2001, the influential researcher Karen Spärck Jones—who began her work in MT at Cambridge, co-publishing a first paper with Masterman in 1958—described the field’s development as “cyclic.” The focus of concentration tends to jerk and loop “because solving problem X depends on solving problem Y but sometimes just because problem Y seems more tractable than problem X, or because there is market interest in solutions to Y.” Spärck Jones charted four phases following the initial period of MT. Roughly, these break down as the semantic phase, the building of knowledge bases; the grammatico-logical phase, concerned with parsing grammar; the statistical processing and probabilistic parsing phase, enabled by the availability of source texts culled from the Internet; and, most recently, a growing concern with practical tasks as demand for language applications increases.

Machine Translation, though still leaky, is now available through search engines like Google. According to arbitrary and completely unscientific tests conducted by me over the past six months, it's basically seaworthy, if you consider an inflatable raft purchased from the seasonal items aisle of an inland grocery store seaworthy. If your expectations are low and you bring enough knowledge of what you're looking at to auto-correct the computer's goofy missteps, you probably won't drown when you use it. (I was requesting translations of restaurant reviews and opening hours, not masterworks of metaphysical philosophy.)

Wilks deems significant recent advances a result of "information compression and minimization of effort." He writes, "All science is information compression, in a wide sense, and it is certainly plausible that the brain, and any other language machine, will have available distinctive procedures to do this, as opposed to the brute force methods of statistics, which are implausible as models of human language processing." There may be a better way to compress (and decompress) information than the way we currently build machines to do it. In a world governed by formal logic, a propensity for ambiguity and illogic is irrelevant; in a world governed by language, ambiguity and illogic are cornerstones of communication. Our imprecision may well be key to our cognition. If that's so, then to make computers more useful as language processors, might we need to make them less perfectly inhuman? Less programmatic? More random? Is that even plausible?

On the question of formalism versus representation, Wilks remains consistent with his earliest conclusions that "purely logical representations, purged of all language-like qualities" are impossible. Words are symbols. But the attempt to rid words of all their representational qualities, while "ultimately doomed to failure … should be pushed as far as it can be, consistently with the intelligibility of the representations we use" for "unrefined representations can be useful, a fact formal critics find hard to understand or explain." Here, Wilks courts Wittgenstein's shadow.

BoTSL#3 2012 APR 15 4:49 PM

In his 1929 *Lecture on Ethics,* Wittgenstein wrote that to use certain words in an absolute sense—to express some ethical, aesthetic, metaphysical value—is but a logically indefensible attempt to exceed significant language. Although this tendency "to run against the boundaries of language," is hopeless, in that the nonsense of our most cherished terms is irresolvable, to charge the cage is perhaps inevitable, and the effort is not expended without consequence. Indeed, in a recent book of Masterman's collected papers, *Language, Cohesion and Form*, Wilks wrote of his colleague, mentor and friend that she

wanted her theories of language to lead to some such goal: one that sought the special nature of the coherence that holds language use together, a coherence not captured as yet by conventional logic or linguistics. Such a goal would also be one that drew natural language and metaphysics together in a way undreamed of by linguistic philosophers, and one in which the solution to problems of language would have profound consequences for the understanding of the world and mind itself. And in that last ... she differed profoundly from Wittgenstein ... who believed that the consequence could only be the insight that there were no solutions to such problems, even in principle.

Suffice to say, Masterman also differed from the surgeons of the Royal College who sought to understand mind as a discernible property of some specific gray matter. Mathematical formalism detaches meaning from form, but in the case of language, the exorcism doesn't kill the ghost. At least, a dualistic, physic-metaphysic metaphor separating the body of formal logic from the spirit of words proves inadequate thus far to the cause of NLP. While it's almost never a good idea to try to explain a joke, whether it is technically possible remains to be seen. Regardless, there will be much to discuss with machines, and computers may soon provide a good enough imitation of life that their consolations will be hard to resist. The question of what is really what or who is really whom will matter more or less then.

*

YORICK'S SKULL.

An awe inspiring but interesting illusion.

"Now get thee to my Lady's chamber, and tell her, let "her paint an inch thick, to this favour she must come."

HAMLET. Act. V.—Scene 1.

DIRECTIONS TO SEE THE GHOST.

Look steadily, in a good light, for thirty seconds at the mark × in the eye of the skull, and then at a sheet of paper, a wall, the ceiling or elsewhere, and continue your gaze fixedly for another thirty seconds when *an awe inspiring and ghost-like skull will slowly appear!*

***By increasing the distance* the apparition will increase in size, so that at five or six feet it will appear of huge proportions.**

PRESENTED BY PEARS' SOAP.