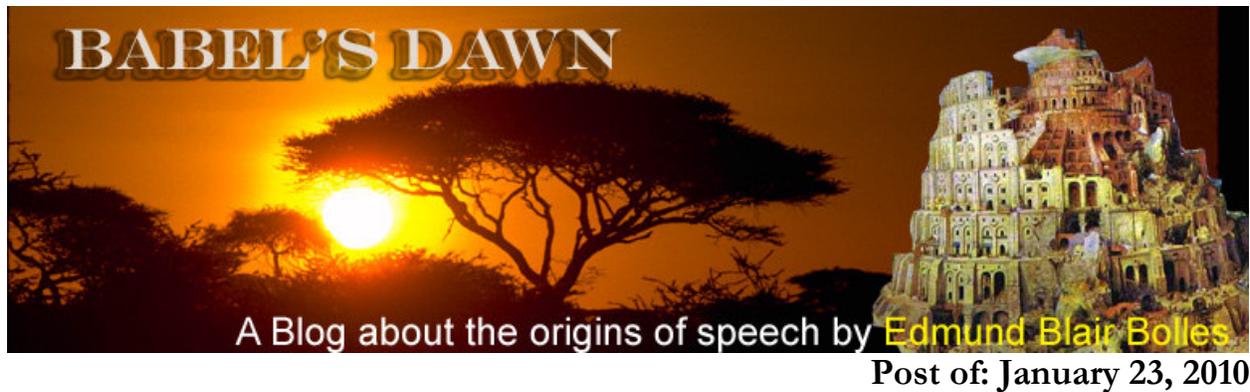
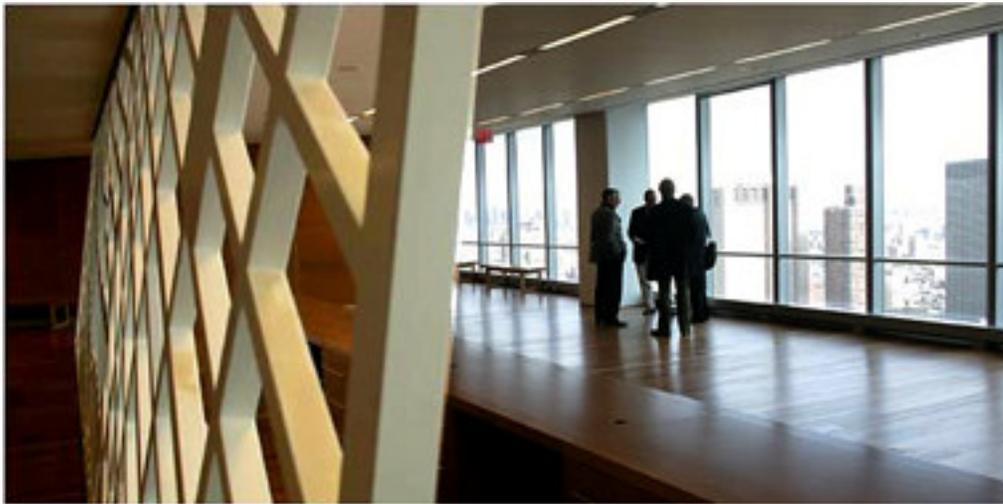


+



Computation Is So Pre-Darwinian



The New York Academy of Sciences recently moved into its new offices offering a spectacular view of Ground Zero and downtown New York. It's a great place to get new ideas.

“Brains don’t compute thoughts, they evolve them,” [Terrence Deacon](#) told an audience at the [New York Academy of Sciences](#) last November at a celebration of the 150th anniversary of the publication *The Origin of Species*. If that claim holds true, then the words we speak are not put together by a computational procedure but emerge from some evolutionary process. Could that be?

Deacon's remark summarized a presentation made that evening by [Gerald Edelman](#), winner of the 1972 Nobel Prize in medicine for his work on the immune system. Edelman's Nobel work showed that the immune system evolves its protection against intruders, and now for many years Edelman has been studying how the brain might work as an evolutionary system. Although I've been generally aware of this research, it took my evening at the New York Academy of Sciences to get me thinking about the differences between usages that are computed from those that are evolved.

Of course, there is the well-known fact that languages do evolve. That might count a point toward the evolutionist's side, but computationists say it doesn't matter because language's underlying universal grammar persists without changing. Computationists acknowledge language undergoes many small changes (microevolution) but they deny that language ever takes on a new form (macroevolution).

What are the differences between something computed and something evolved?

Predictability v Possibility

Here is the basic distinction that draws me toward the evolutionary side of this comparison. Edelman's idea implies an answer to what has always struck me as language's central mystery: its creativity—a capacity to generate original, apt remarks.

Computation is predictable, reliable, repeatable. The results are the same, no matter who or what does the computing, so long as the computational rules are followed correctly. Evolution is none of those things. It is predictable only in a general sense; e.g., if you become infected with an H1N1 virus, your immune system will respond and probably produce an antivirus. We can't predict exactly what that antivirus will be like, or whether it will come in time to prevent the virus from killing you, or how the H1N1 virus itself will evolve to get past those antiviruses in the future. This unreliability might make computation seem the safer route, but a computation system cannot protect against unprecedented dangers. When the environment is unpredictable, predictable solutions will not always serve. Whenever the old rules no longer work, evolution's ignorance of rules provides a path to the new.

Purity v Entanglement

Computation treats the environment as irrelevant to the procedure, while the environment is fundamental to evolution. A computational device can easily be located on a shelf in a sealed room, processing data and generating outputs to an internal log. As long as it has a set of rules to follow, it can function. The constraints on performance all have to do with the machine's specifications. But an evolutionary system can only work when it is part of something larger. Its constraints come as much from the environment as from its own structure.

On this blog I have said many times that language depends on a triangle in which speaker and listener pay joint attention to a topic. I never thought about it before this post, but plainly this arrangement is evolutionary rather than computational because it depends as much on the listener and the topic as it does the speaker. A staunch computationalist like [Noam Chomsky](#) denies the importance of this triangle, or even that language is primarily a communication system. But computationalists have never found a solution to the problem of meaning. (See last week's post, [The Quest for Consensus](#), where [Tecumseh Fitch](#) bemoans the mysteries of meaning.) Until computationalists do find some persuasive way of explaining meaning without referring to the environment, I'm chalking up this round to the evolutionists.

Rules v Reasons

Computation is rule-based; evolution is *ad hoc*. Language gets two very different portraits, depending on which approach a person follows.

Chomsky is the star computation proponent and he says that language can produce an infinite number of sentences because the rules allow for infinite recursion. Thus, a finite set of rules allows for an infinite variety of sentences.

The evolutionist, on the other hand, may agree that speech is unbounded, but that's because there is no cap on what might emerge, just as there is no limit on the forms that can be evolved by life on earth.

It is hard to tell if any particular output comes from a rule or a process because we can always describe a rule for any particular utterance. The challenge comes in finding rules that account for the whole of language. The same problem faced the pre-Darwinian biologists who were looking for mechanical laws that would explain life forms. Naturalists like the Baron Cuvier and Louis Agassiz eagerly studied the rules that governed zoological forms. Their work was overthrown when

Darwin came along and said there are no such laws, something else is afoot. Will that same thing happen to modern linguists?

Centralized v Scattered

Computational rules are hierarchical; evolutionary processes are localized. Even if a computation uses parallel processors, the results of those separate operations are bound together in some central unit. Evolution acts on the spot without any other controls on output.

This difference seems to look promising for the computationalists who have developed hierarchical rules for structuring sentences. But neurologists have found that the formation of a sentence comes from points scattered across the brain. Evolutionists will have to develop an account of sentence emergence that does not follow hierarchical rules, or computationalists will have to explain the neurological data. So far, this point looks like a stalemate.

Friction v Novelty

Computation treats variation in output as a friction on performance, while evolution treats it as the stuff of new expression. If a speaker says, "I an apple see," a computationist will say the speaker has made a slight mistake and chalk it up to something going wrong between the sentence's computation and its output. An alert listener, however, might find a certain charm in the phrase, liking the accent it puts on the object and think to use it later when the moment seems apt. If the listener does use the same structure later, a computationist will chalk it up as another bit of friction in performance while an evolutionist sees a new type of expression taking hold.

Intelligent Design v Selection

In the end, a computation system is only as good as its design. A designer might include room for some feedback. For example, a talking computer can ask, "Is your name Arthur?" and, if told no, might generate a new sentence, "Is your name Paul?" But such feedback loops have to be designed into the system. Meanwhile, a talking evolutionary system will say, "Hi, Arthur. Oops, I mean Paul." That last little business, examining the output, judging and replacing it is typical of evolutionary systems. Evolution is only possible if there is a way of favoring apt outputs over others.

Our ability to talk and think usefully about new things has always been a problem for computationalists. The philosopher [Jerry Fodor](#) insists, as [Leibnitz](#) did, that everything we talk about is innate, and if evolution cannot explain how we happen to have an innate concept of iPods, so much the worse for evolution. Most computationalists are less radical than that, but for all of them the ability to compute correct, unprecedented solutions to unprecedented problems is a mystery. Certainly computers cannot do it without getting an upgrade.

For evolutionists, this distinction at long lasts suggests a biological function for awareness. It offers a way to immediately correct an inapt output. For example, if I address Paul as "Arthur," and pay attention to what I'm saying, I can notice the mismatch between the word spoken and the fact before me. Attentive selection can work much faster than natural selection, and the speed of language evolution is several magnitudes of order greater than biological evolution.

So do we compute or evolve our words?

Links:

Terrence Deacon: <http://anthropology.berkeley.edu/deacon.html>

New York Academy of Sciences: <http://www.nyas.org/>

Gerald Edelman: http://nobelprize.org/nobel_prizes/medicine/laureates/1972/edelman-bio.html

Noam Chomsky: <http://www.chomsky.info/>

The Quest for Consensus: http://www.babelsdawn.com/babels_dawn/2010/01/the-quest-for-a-consensus.html

Tecumseh Fitch: <http://www.st-andrews.ac.uk/~wtsf/>

Jerry Fodor: http://en.wikipedia.org/wiki/Jerry_Fodor

Leibnitz: http://www.maths.tcd.ie/pub/HistMath/People/Leibniz/RouseBall/RB_Leibnitz.html