A.L.I.C.E. AI Foundation

Promoting the adoption and development of Alicebot and AIML free software.

Why You *Don't* Need Proprietary Bot Software

Noel Bush
noel@alicebot.org

Windows 0.3

In the year 2001, there is no HAL 9000. Instead, there are a growing number of software packages that attempt to do some part of what Arthur Clarke / Stanley Kubrick envisioned in the book and film of the same name, most of which go by the label "chatter bot", "virtual representative", "virtual person", "smart bot", or some similar variant. Here I'll be calling them all "bot software", disregarding the fact that "bot" usually applies to a broader range of software than just conversational programs.

Many people claim that this bot software has real business applications in customer service, distance learning, and so on. A number of companies have been built in hopes of capturing a piece of a still infant market for bot software.

Most of today's bot software is proprietary, meaning that its source code is closed, its methods layered over with some attempts at legal protection (patents, etc.), and its use restricted to those who agree to pay license fees that are usually very large, and that are usually followed up by hefty professional services fees.

Only a few bot software packages are free or open source, and only one has had the participation of over 300 developers around the world. That one is A.L.I.C.E., today known as the Alicebot engine and AIML technologies. Alicebot/AIML is released under the GNU General Public License, the same license used by Linux, and the truest of "open source" licenses in that it construes a particular meaning of "free" that has to do not with price, but with access and liberty. (See www.fsf.org.)

Alicebot/AIML is to [insert proprietary bot software] as Linux 0.01 would have been to Windows 0.3. Of course, when Linux 0.01 was released (1991), Windows 3.0 was already out, introducing such marvels as the use of more than 640Kb of RAM; and of course, no one but an inner circle at Microsoft ever saw such a thing as "Windows 0.3" (if there was such a thing).

But there are two points to this analogy:

1. Alicebot/AIML is already more advanced than any commercial bot software.

2. Commercial bot software already faces the same uphill fight against the market forces of free/open source software that Microsoft did not begin to face seriously

until its own software had advanced significantly beyond an embryonic stage (say, around 1999 when Red Hat went public and Windows 2000 was released).

Add to that that most commercial attempts to sell proprietary bot software are struggling to stay alive, and that the list of Alicebot/AIML companies is growing. Below I'll go into some detail to explain why you don't need—and shouldn't choose—proprietary bot software if you are planning to begin using bot software in your company. This argument is divided into three parts:

- The Historical Case
- The Technology Case
- The Business Case

Feel free to skip around.

The Historical Case

A bit of a history lesson is in order. Some bot software companies try to place themselves in an historical timeline that usually starts with Joseph Weizenbaum's ELIZA. They provide a story something along these lines: ELIZA was a primitive attempt at conversational AI that sparked imagination, but mostly relied upon a few simple "tricks"—and since then, some geniuses have been locked away in an unmarked building working up a true revolution in 'artificial intelligence', which just so happens to have landed on the doorstep of [company name].

The idea is supposed to be that ELIZA whetted the world's appetite for talking computers, but that the approach used wasn't serious enough, and that it took a few more decades for "real" NLP (natural language processing) to fill in the missing pieces and make today's bots able to carry on remarkable conversations, sell products to customers, remember web site visitors, teach about subjects, give financial advice, and so on.

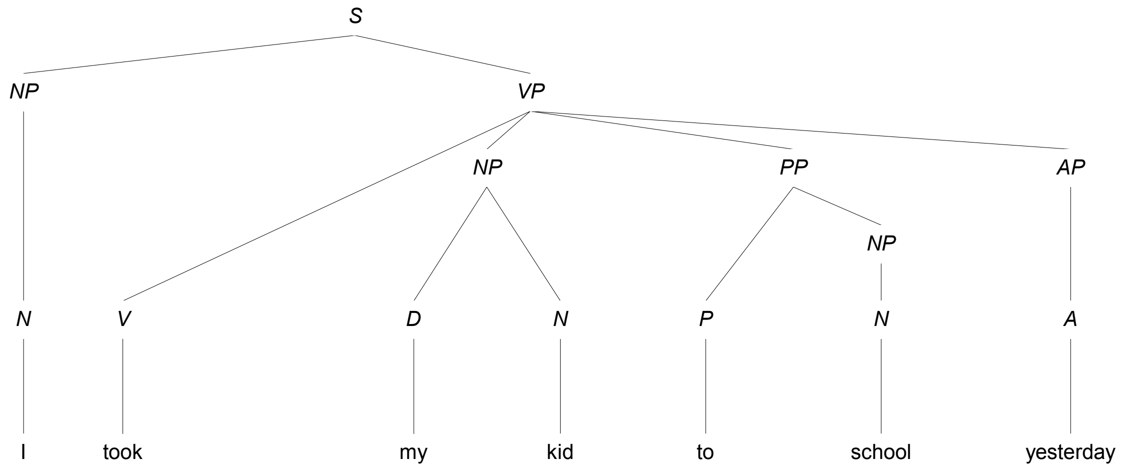
But the fact is that the development of what most people call "real" NLP and the development of bots have mostly proceeded along different lines. "Real" NLP as it's understood in the academic world goes deep into problems of linguistics that are still poorly understood. Most of the work that has been done in conventional NLP has not translated into something that can be used for making a machine carry on a better conversation. In fact, from the point of view of academia, the software that's advertised as conversational bot software mostly relies on "outdated" ideas that were discredited almost as soon as they were introduced.

Richard Wallace is the only (former) academic who makes a serious case for the legitimacy of the approach used in ELIZA, and has also demonstrated the possibility for expanding on that approach to produce results that match what people are looking for in bot technologies. Commercial companies using closed software generally use talk of patents and advanced secretive technologies as marketing tools to mask the truth—that they too are relying on an approach that is also heavily based on the original ideas of ELIZA.

In truth it's the conventional NLP approach that is stuck. Any professional linguist will tell you that for as many advances as the study of language has made, it has encountered as many profound setbacks. Most conventional NLP approaches rely upon fitting actual speech to a meta-model of "meaning" and communication. Briefly, for instance:

"I took my kid to school yesterday."

might be illustrated as:



The sentence is broken down into components based on their "syntactic" function. This syntactic analysis is meant to serve as a framework for understanding the "semantics" (meaning) of the sentence. A machine needs to know things like:

- "took" is the simple past tense form of "to take"
- "I" refers to the speaker of the sentence
- "my" refers to something that belongs to the speaker of the sentence
- "kid" can mean many things, among which is a "child"
- A child is a young person
- "my kid" probably indicates that the speaker is the "parent" of the child
- "parent" can mean many things, among which is a person who is somehow responsible for another person, possibly including that other person's upbringing, birth, etc.
- "school" is a place where people learn
- "to take" someone "to school" means to arrange for the transportation of that someone to the school and to accompany them (i.e., "to take" alone is ambiguous)

And so on. There are many theoretical approaches to filling out the details in this fashion. Many approaches take a starting point like this, and rely, at some level, on several important resources:

- a "lexicon" of words that describes their different forms and possible positions in a "sentence"
- a "corpus" of semantic information about different combinations of lexical items and the relationships of those combinations to other combinations
- an engine that can apply a lexicon to a sentence and identify all possible interpretations of the sentence's syntactic structure
- an engine that can apply a semantic corpus to a series of syntactic & morphological ("word shape") parses and identify the most likely interpretation of the semantic intent of that sentence

And in fact the notion of what constitutes a "sentence" is also poorly understood, believe it or not, leading some people to speak of "utterances" rather than sentences.

It's also clear that even a good analysis of a sentence in isolation is relatively useless, leading to the need for "pragmatics", in which the analysis attempts to locate the meaning of an utterance within its "layers of context". The notion of "context" is also poorly understood, and at its best in computer NLP is treated as a sort of giant static network of "meanings", often called an "ontology".

There are a huge number of theories of NLP that follow an approach that looks something like this. None of these approaches have a complete set of tools that can handle real text generated by humans, to consistently produce an accurate analysis with which most people would agree.

There are also approaches that try to minimize the number of a priori rules, and rely instead on "learning" techniques based on statistical analyses of large bodies of text. Neural networks, fuzzy logic, genetic algorithms and other approaches are essentially statistically-oriented techniques that try to remove some of the manual labor from statistical analysis by introducing a "black box" that iteratively builds rules, often hidden from human users, based on feedback about the analysis from human controllers and/or "heuristics" that serve as non-absolute guides to analysis.

Aside from the fact that no approach fully satisfies its own stated aims within whatever constraints on language and goals for analysis are set, almost all approaches suffer from an inability to "explain" the workings of more than a handful of human languages with the same set of rules. And virtually no approach can do more than acknowledge that even a single given language, say English, is used in countless "standard" ways and evolves continuously as it is used.

One only needs to begin a web search on "natural language processing" to understand that the amount of research into this topic is immense, and that no approach has a legitimate claim on being the best. The history of linguistics is rife with fierce battles among academics whose entire careers have been staked on establishing the authority of one approach over others—none has met with success (see Randy Allen Harris's *The Linguistics Wars*[1] for a great read on this topic).

Suffice it to say that the academic debate and research struggle forges ahead vigorously, but that none of its output has proven commercially viable.

Perhaps the most spectacular failure has been the CYC project, initiated by Doug Lenat in 1984 and beneficiary of millions of dollars of government research money and private and institutional investment. CYC (now housed under a company called Cycorp[2]), was and is a project with the aim of building a giant knowledge base full of "common sense", with the idea that this would someday enable machine understanding of texts. Lenat has been telling journalists (and presumably his investors) for years that CYC is mere months away from being able to understand simple texts like *TIME* magazine. So far quite a lot of money has been spent, and the giant knowledge base continues to grow, but despite its intricacy (some might say beauty), its commercial use is still beyond reach, and its theoretical base lags behind the academic research front, itself still light-years away from success.

Know this: the commercial bot companies may borrow bits and pieces from conventional NLP, but by and large they are every bit as ELIZA as ELIZA. As ELIZA relies on pattern-matching and simple string manipulation, so too do all "proprietary" offerings at their core.

The Technology Case

Given all that, you might still reasonably assume that behind locked doors some innovation has taken place, funded by visionary investors and executed by brilliant developers, that justifies offering a proprietary bot technology at license prices in the hundreds of thousands or millions of dollars (US).

You might reasonably assume that, but you would be wrong.

A.L.I.C.E. has been around since 1995, before most of the current crop of commercial bot companies even began. Richard Wallace is very forthright in describing his invention of A.L.I.C.E. as based directly on insights learned from ELIZA and its various successors. As the original documentation for A.L.I.C.E.[3] shows, A.L.I.C.E. was an evolutionary step that modularized the simple string manipulation and memory techniques used by ELIZA and allowed the rule base to be easily extended to perform one of a limited set of operations on a client input, producing an output that incorporated information (1) already in the bot's knowledge base, (2) "learned" from the client, and (3) derived from context.

Nothing was ever hidden in the A.L.I.C.E. approach, and given the timeline and the fact that A.L.I.C.E. was always under the GNU GPL, one might make a reasonable case that the bulk of the approaches used by proprietary bot technologies probably violate copyright. Fortunately for vendors of proprietary bot technologies, the aim of Wallace's use of the GPL was never to lay claim to personal proprietorship of the approaches used in A.L.I.C.E., but rather to encourage innovation by opening up the project to as many participants as possible.

[1] http://www.oup-usa.org/isbn/019509834X.html

[2] http://www.cycorp.com/

[3] http://birch.eecs.lehigh.edu/alice/readme.html

Based purely on interactions with online demonstrations (and customer implementations) of bots produced by proprietary vendors, one may very reasonably challenge any claim of technological superiority of proprietary technologies over Alicebot/AIML (or any other free/open source bot technology).

You don't have to believe me, of course—try it for yourself. Here are some things to watch for:

- Can you "trip up" the bot using sentences that reveal its primary reliance on keywords?

  For example, try asking a bot a question that uses two or more important words, like: "What is the difference between natural language processing and professional cosmetology?" These are the answers I got from several bots online:

answer	Bot Name	Vendor	Company Using It	URL
"I'm not sure what you are asking me. Part of my brain is a proprietary natural language processing engine. I hope that helps answer your question."	Julia	Virtual Personalities, Inc.	Virtual Personalities, Inc.	http://www.vperson.com/
"Natural Language refers to a computer's ability to understand the things a human says, without the human having to talk using special codes like programming languages."	Nicole	NativeMinds, Inc.	NativeMinds, Inc.	http://www.nativeminds.com/
"I am sorry, I did not understand your question. Below are some topics that people often ask about. You may also try rephrasing your question."	Hank	NativeMinds, Inc.	The Coca-Cola Company	http://www.cocacola.com/
"'Natural language' is what you speak and hear all day."	Karl von Wendt	Kiwilogic.com AG	Kiwilogic.com AG	http://www.kiwilogic.com/
"I'm not quite sure what you mean."	Liam	Kiwilogic.com AG	Loewe AG	http://www.loewe.de/cda/frameset/0,2206,ge-en-fl11-0,00.html
"It is just computer software trying to understand and generate human language."	Luci	Artificial Life, Inc.	Artificial Life, Inc.	http://www.artificial-life.com/
"I do not have an answer for you. But in the meantime, let's move on. And let me ask you, which type of financial goal are you considering the most: retirement; estate planning; college; or other?"	Janet	Artificial Life, Inc.	Pioneer Investments	http://www.pioneerfunds.com/

A.L.I.C.E. herself answers that question, "Aren't they the same?"

What you should notice about all of the responses from the proprietary bots is that they either (a) pick out one key word/phrase ("natural language processing") and provide a quick description of NLP, ignoring the rest, or (b) flat-out admit that they don't know the answer, and try to steer the conversation to topics they do know. In no case do they even acknowledge that you've asked about the difference between two things, although that certainly doesn't need a grammatical parser to detect.

The only difference in A.L.I.C.E.'s response is that her knowledge base contains the pattern "WHAT IS THE DIFFERENCE BETWEEN *" and the witty come-back "Aren't they the same?"—basically a gamble on either amusing the client or occasionally hitting the right answer when someone tries to trick the bot by asking what the difference is between two things that are, in fact, the same.

- How well does the bot really remember the thread of your conversation? Many bots will ask your name, and incorporate this into future responses. Some bots also attempt to identify the "topic" of conversation, and a few will be able to tell you what the topic is when you ask about it (although even this is rare).

  Just remember that asking your name, or any other piece of information, and storing it in a variable for later use doesn't constitute "understanding" what's going on in any sophisticated sense, nor is it the basis for "reasoning" in any sense of the word. Many companies try to pass off tricks like this as a kind of "memory" or "knowledge of context".

- How consistent (if present at all) is the bot's understanding of "semantic relations"? Many—though not all—theories of language include the notion that the things named by words are related to one another in a myriad of (usually uncountable) ways: for instance, a Saab is a kind of car, a car has wheels, a car is a kind of vehicle, etc. Some bots may include a tiny, domain-specific set of apparent "knowledge" about these relationships. For instance, a bot selling a product may be able to answer the question "How much does it cost?". But if you continue asking several questions about the product's attributes and relationships to other products and "things in the world", you'll soon see that the apparent knowledge is a sham.

- How well does the bot really understand "anaphora"? Anaphora are references to previously-named things. In English, pronouns like "he", "she", "it", and so on usually refer to previously named people or things. Some bot companies will pretend that their software can really identify the target of an anaphoric reference. But such claims can be proven false, and can easily be shown to be keyword-based as well. You need only pursue a dialog a few turns to show that a bot really has no clue what you're talking about, beyond a rudimentary index of the "current topic".

You can try all the tricks you want; some of them will get you witty answers, a few will even be answered "correctly" (until you twist the trick a bit more), and most will just get you a plain stupid answer. It's probably also worth pointing out that any of the bot companies that are still functioning and happen to come across this article may improve their response to the example question above.

The point is that the Alicebot/AIML *technology set* is no more or less sophisticated in its ability to handle these kinds of tricks than any proprietary technology. That's because answering "trick questions", depending on whom you ask, (a) depends on solving deep NLP problems that are still plaguing us, (b) just depends on a good "botmaster" who reviews conversation logs and continually improves the knowledge base so that the bot answers the question better the second time around, or (c) is plain impossible.

The A.L.I.C.E. theory is that (b) is the best answer for now, and that this is perfectly adequate for most realistic uses of bots. That might sound like a cop-out, but it's honest. It's more honest than any claims of sophisticated "reasoning" or "discourse management" capabilities that proprietary vendors might make. As Wallace says, trick questions (and there are entire corpuses of these in any standard linguistics text), "reveal the ambiguity of language and the limits of an ALICE-style approach (though not these particular examples, of course, ALICE already knows about them)."

So proprietary vendors tend to shy away from emphasizing their claims of NLP superiority any more than they feel they must in justifying the "sophistication" (and consequent high price) of their technology. Any company that decides to invest in bot technology and goes through a process of technical due diligence in comparing bot companies to one another will quickly get into a discussion of more run-of-the-mill issues that apply to implementing any technology in a web environment (or otherwise high-use situation): things like scalability, integration capabilities, and the development infrastructure behind the software itself.

Scalability: as of this writing, the Alicebot/AIML technology set as a whole hasn't undergone any massive testing for load handling, fault tolerance, and so on. But the "Graphmaster", which lies at the heart of the Alicebot engine, does have a mathematically demonstrable scaling factor. Wallace explains it as follows:

> The Graphmaster is a directed tree graph, with a root node, and pathways from the root to the terminal nodes. The terminal nodes contain maps to the <template>, <filename> and potentially other properties of the category. Each terminal node is one category, and the number of terminal nodes equals the number of categories.
>
> The basic efficiency of the Graphmaster comes from the observation that, given a sample of sentences of natural language, the branching factor for the first word is large (around 2000 in our case), but the average branching factor for subsequent words of a sentence decreases rapidly (e.g. 2000,3,2,1,1,1,...) So the tree for natural language sentences is shallow, but wide. This observation does not depend on whether the basic tokens are words, or letters (I've implemented both types of Graphmasters), or whether the input set of sentences is a corpus of New York Times text, the Bible, a collection of e-mail bodies, or the Pattern set of A.L.I.C.E.'s brain.
>
> We have actually plotted a collection of Graphmasters for the ALICE brain. The pictures are quite beautiful. See http://www.alicebot.org/documentation/gallery/.
>
> If the number of categories is N and the average pattern length is M, the memory required to store them in an array (as in program B) is O(MN). But the Graphmaster compresses common prefixes together, so requires much less than MN.
>
> [About "Big O" notation: In general O(f(n)) means there exists constants k and c such that for every input problem larger than cn, the computation takes less than k*f(n) steps. O(1) means that the time is constant and does not depend on n.]
>
> Consider the case of an "atomic" pattern (no wildcards). The time to match an input to that pattern depends only on the path length from the root to the corresponding terminal node. This takes, on average, M steps, irrespective of the number N of categories. This is why we say that matching in the Graphmaster is O(1). At least one of our competitors advertises a matching algorithm that is at best O(log N). [As an exercise--prove that the time to match even with wildcards does not depend on N.]
>
> The Graphmaster is implemented elegantly with the Java Collections Library. Essentially each node is a map, called a Nodemapper, which takes a given node, the next word in a sentence, and returns the next node down the path, or null. In practical terms, the recognition time is around 10ms, no matter how many categories are loaded, up to the limit of memory. The response time can be slower because of <srai> chains and <template> evaluation, but our Graphmaster is the most efficient structure, in terms of both time and memory, that one can imagine for solving our matching problem.[4]

You won't find a more detailed explanation from any vendor of proprietary bot technology. The scalability of the Alicebot engine itself is open to the world for inspection and verification. Few, if any, proprietary vendors can answer this type of question in any other way than to describe aspects of their *server*. You may get answers describing how many simultaneous connections the *server* can handle, or what kinds of fault tolerance and

[4] from the A.L.I.C.E. and AIML Mailing List: see http://alicebot.org/mailingLists.html#general

rollover capabilities the *server* offers, but press for answers about how the knowledge base itself scales and you'll likely hit the "proprietary techniques" brick wall faster than you can say "How's your stock trading today?".

With respect to server scaling, the latest Alicebot implementation—called "Program D" and also known as "Alicebot.NET"—is built to use "enterprise-aware" technologies like the Jakarta Tomcat application server[5] from Apache (also open source) as well as the Apache server itself, and the open source Hypersonic SQL database system[6] (as an example, although any JDBC-compatible database can be swapped in). Not only does the commitment to the use of free/open source technologies mean that important enterprise scaling considerations can be investigated fully, but the use of J2EE-friendly technologies[7] means that the Alicebot/AIML technology set can be counted on to fit well in a modern enterprise application environment.

Claims of better scalability or intergratability from proprietary vendors simply cannot compete with a free/open source project that exposes its full workings to the world, and receives scrutiny from an impressive roster[8] of individuals from around the world and from a variety of technical, managerial, and expertise backgrounds.

One last thing I've got to include here is the fact that Steven Spielberg's web site for the movie AI[9] has been using A.L.I.C.E. for the last several months. The Spielberg people aren't talking, and in fact never contacted Richard Wallace or anyone at the Foundation about their use of the technology (as Wallace says, that's "the magic of open source at work"), but it's interesting to note that a million-dollar viral marketing campaign that seeks to draw as much traffic as possible to its network of over 70 sites, with no direct sales through the sites—just pure entertainment—chose the Alicebot engine over any commercial variants. "We are selling movie tickets," says Wallace.

Strip the proprietary vendors of their claims (or hints) at better technology, either in the core engine itself or in the "enterprise framework", and you're left with little justification for investing. But of course there's one more piece—the business case—that proprietary vendors will try to sell you. It's equally flimsy.

The Business Case

Given whatever technological or theoretical argument you may raise, any half-awake vendor of proprietary bot technology will come to its own rescue by promoting the business sense of dealing with a product developed, maintained and supported by a commercial entity. In truth, there is business sense in having an entity that's "behind it all". That's part of why we created the A.L.I.C.E. AI Foundation, to provide a much-needed resource for companies that want to use the Alicebot engine and AIML technologies.

Unfortunately, a commercial entity like one of the proprietary bot vendors doesn't provide a reliable resource. Proprietary software can only be supported to the extent that its owner is able to function. Most bot companies relied on the overhyped technology market

[5] http://jakarta.apache.org/tomcat/

[6] http://hsqldb.sourceforge.net/

[7] http://java.sun.com/j2ee/

[8] http://alicebot.org/committees/architecture.html

[9] http://aimovie.warnerbros.com/

of the late nineties to make their start; since the end of that boom, the chips are falling and companies once again have to prove the actual soundness of their business models. Some are making it, some aren't.

What happens when a vendor of proprietary software goes bust? You lose your support. You lose your ability to add new functionality. You basically lose the whole investment you made in license fees, implementation, and professional services from the vendor. Unless you were persistent and smart enough to secure rights to the source code that would at least give you some deeper access to the technology in the event of a financial catastrophe, you are simply out of luck.

All the claims of proprietary algorithms, all the professional services that may be on hand, all the development power that may be exhibited as being available to customers of a proprietary vendor, don't add up to anything except money thrown out the window if that vendor hits hard times.

And what if it's Microsoft that is selling the technology? They aren't yet, but they have a research lab[10] full of brilliant people trying to solve these problems (among others). Would you buy it if it came from Microsoft?

Well, had A.L.I.C.E. never existed, you might. But return to the analogy I started with. Compared to proprietary bot technologies, Alicebot/AIML is at least as advanced, and likely moreso. Since proprietary products are by their nature shrouded in secrecy, and since bot technology especially is something where a lot of smoke and mirrors is necessary to hide the relative lack of differentiation among solutions on offer, making that comparison is, naturally, a matter of some detective work.

You might know that I used to work for one of these proprietary vendors[11], and you might assume that I'm hinting at some "secret knowledge" about this field—in fact, I'm not. I'm here as a convert from proprietary approaches, telling you that what you can see in Alicebot/AIML stands on its own against any commercial competitors. The fact that A.L.I.C.E. beat out all other competitors, including the product from my former company, at last year's Loebner Prize[12] is a nice testament to the quality of this technology…but as I say, you can see for yourself.

The fact is that even if (all right, *when*) Microsoft or another giant releases a conversational bot engine, the world today is different from when Microsoft introduced Windows. Back in 1985, Windows 1.01 was released and a scruffy Bill Gates[13] described it to the world as a "revolutionary concept in software"—evidently enough people agreed with him. But there wasn't a better solution already brewing. Apple's new OS was perhaps more elegant in its graphic user interface, but arguably even more closed than Microsoft's, due to its reliance on proprietary hardware. Microsoft was the company that demonstrated to the world—specifically the business world—the need for a business-oriented operating system with a graphical user interface. Even Linus Torvalds's release of the earliest version of the Linux kernel wasn't complemented by a usable GUI for several years. And performance and scalability aside, it's the GUIs like GNOME and KDE that have really begun to break down the doors for Linux in the corporate world.

[10] http://research.microsoft.com/

[11] http://www.artificial-life.com/

[12] http://www.dartmouth.edu/~phil/events/Contest_Results.html

[13] http://www.mindspring.com/~jforbes2/winhist/billgates.jpg

Today, even a significantly smarter conversational bot engine released under proprietary auspices will have a hard time competing with and staying ahead of the Alicebot engine. Any advance will either be based on academic research, which always comes to light, or on a better arsenal of string-manipulation tricks, which can always be duplicated without looking at one line of source code.

Any proprietary offering will have the added burden of needing to maintain and support its product, whereas a free/open source project receives development support from an unlimited community of interested parties. The ability to support a free/open source technology waxes and wanes with the market demand for the technology. The ability to support a proprietary technology may or may not increase with the financial health of its vendor, and most certainly dies along with its vendor.

In cold hard numbers, a company seeking bot software for its web site, internal network, or whatever other use, simply cannot justify *not* using a free/open source solution. Paying $50,000, $100,000, $200,000, or millions of dollars simply for a license, and then adding onto that an indefinite need for support from an exclusive service provider, does not a reasonable decision make.

Paying $0 for software, and possibly only incurring the costs for internal expertise to install and customize it, makes a lot of sense. Contracting an independent service vendor (or individual), and having the freedom to change that vendor at any time, is not merely "nice" but the only strategy that makes sense.

Look at the list of companies that are providing Alicebot/AIML services[14]. Today, to be honest, most of these companies are limited in their descriptions of services provided, and limited in available references. Tomorrow that will change, you may be sure. And given that the licensing deals by proprietary vendors and their big-name clients are almost always secretive, you have no way of knowing whether the clients currently listed by *those* companies (at least those who list clients) are in fact companies that paid a regular fee, or whether they negotiated reduced fees that the vendors accepted in order to "get out the name" (or pay the bills).

The market is simply too young to measure at all. What isn't difficult to see is that even the gap between a publicly-traded company and a couple of individuals operating down in Atlanta isn't as big as one might think, given that the first may have (or may have had) a few millions of dollars to throw around, but the second has at its disposal a technology that has probably had more development effort put into it than all the commercial variants combined.

Last, a few words about "market research". Bot vendors sometimes try to bolster their claims with material adapted from market research companies. There is, for example, a tired old Forrester report that has been re-used over and over again in public presentations and marketing materials by vendors of proprietary technologies, which describes the "cost of customer service" using various traditional methods (phone support center, email), and claims to show how "automation" can save the day. The proposed leap, by the bot vendors, is to suggest that bots can dramatically decrease the cost by eliminating the human operator, or by (in one company's terminology) acting as "tier zero" support. That general *possibility* is, of course, the premise upon which a recent press release from the A.L.I.C.E. AI Foundation is based ("Bots Will Unemploy You"[15]).

[14] http://alicebot.org/companies.html

[15] http://alicebot.org/press_releases/2001/unemployment.html

But it is indeed a big leap from this kind of speculation, which is certainly relevant but also certainly not proven, and any claim that such costs as are claimed can actually be recovered in this way, using any technology available today. I personally like Richard Wallace's point of view[16] that the "call-center mentality" needs to change before these bots will really find their value.

So I encourage you to look with great suspicion on any report that claims to provide real statistics on the market impact of bots. It is clear that these kinds of technology fascinate people immensely, and that that's their main value in the market right now. Fascination counts for a lot. The kind of "service" they can provide is indeed useful and has value, but is not enough to account for the prices at which proprietary "solutions" are sold.

It should be apparent that AIML/Alicebot provides the best and best-understood technology available for rapidly building a bot that can handle many of the sorts of "tier zero" issues that some proprietary vendors have identified. But it's not at all apparent that that translates into an excuse for selling it or similar technology at movie-budget prices. Unfortunately, most of the market information you're bound to find will attempt to distort that fact, and make claims about the capabilities and value of this kind of technology that are far overstated.

Ladies and gentlemen, please…don't pay a lot for that bot.

About The Author

Noel Bush is a co-founder of the A.L.I.C.E. AI Foundation[17] and a member of the Board of Trustees. He is also co-founder of X-31[18]. His career has meandered between music-theoretic/-compositional activities and technological projects. He studied with Benjamin Boretz in the controversial Music Program Zero at Bard College, and most recently was Vice President of Engineering at Artificial Life, Inc.

At Artificial Life, Bush also held at various times the positions of CTO ALife Russia, Vice President of Technology Development, and Product Manager.

Prior to Artificial Life, Bush worked at The Condé Nast Publications, Inc., where he developed helpdesk and knowledge base systems.

As Managing Editor of Open Space Publications from 1993-1996, Bush edited and/or contributed to the production of a number of important works, including the re-issue of Benjamin Boretz's Meta-Variations and J.K. Randall's Compose Yourself.

Bush's essay about and computer program for cybernetic co-composition of music using the Csound language were published by MIT Press. His *intract*, a computer-synthesized work, was recently featured in the Opera Prima Europa in Rome.

Noel Bush was born in Louisville, Kentucky (USA) in 1974. He earned a B.A. in Music at Bard College in 1996. He currently resides in St. Petersburg, Russia.

[16] http://alicebot.org/articles/bush/wallaceConversation.html

[17] http://alicebot.org/

[18] http://x-31.com/