

# Chapter 11

## Wolfram|Alpha: A Computational Knowledge “Search” Engine

John B. Cassel

### 11.1 Introduction

Wolfram|Alpha (W|A) is a search engine in some senses, but not in others. People do use W|A to search for responses to their questions. Additionally, W|A often runs queries, or searches, through its databases. However, what W|A does not search for information containing words that match or are similar to a provided phrase. Instead, Wolfram|Alpha makes new contact with the history of librarianship, using the curation work of library professionals and domain experts to map words into functions describing the query, attempting to encode the question’s intent.

This chapter will first describe how Wolfram|Alpha is different from other search and answer technologies. Next, we will explain how Wolfram|Alpha took this different path, one against the grain of technical trends but clearly sensible when contextualized by the internal forces at play in the Wolfram group of companies. Then we will examine a variety of Wolfram|Alpha queries, demonstrating what this particular approach gives its users. From there, we will talk about the ongoing legacy and the future directions for Wolfram|Alpha. Finally, we will conclude by summing up how this particular technical organization serves people who need to know not what is out there on the internet, but facts that can be readily calculated and summarized when content is carefully distilled.

---

J.B. Cassel (✉)  
Agrible, Champaign, USA  
e-mail: john.benjamin.cassel@gmail.com

## 11.2 Against the Grain: Comparing Wolfram|Alpha's Technology

A key factor in the success of Wolfram|Alpha was not only the quality of its technology, but that it went against the grain of the received wisdom of the time. W|A was created in the height of the Web 2.0 boom when crowdsourcing and collective intelligence were all the rage, but W|A worked by hiring in-house curators who were librarians and domain experts to vet their data. In a time where statistical machine-learning methods were dominating artificial-intelligence development, novel syntactic approaches did Alpha's heavy lifting. It was not information retrieval technology that mined text of the Internet, but instead a symbolic approach that found the computable meaning of queries. Instead of being a product of Silicon Valley, it was a product of a low-key Midwestern town known for its high-tech university and highly-ranked library school: Champaign, Illinois. By going against the popular tide, Wolfram|Alpha had blank canvas for new technological innovation. Let's look into these differences by comparing Wolfram|Alpha with Google, IBM's Watson, and Wikipedia, and then talk a little about Wolfram|Alpha's disparities with contemporary artificial-intelligence trends.

## 11.3 Direct Differences: Wolfram|Alpha in Comparison with Google

It is easiest to see Wolfram|Alpha's difference by comparing it with Google as Google was at the time. The most obvious difference is in the change of Google's results. Figure 11.1 displays sections of what the Wolfram|Alpha result for George Washington looks like as of this writing.

As of this writing, if you search for "George Washington" on Google, you will get the following summary on the right-hand side (see Fig. 11.2):

When Wolfram|Alpha came out, this was not a feature of Google, which had links to other pages, not content in itself. Wolfram|Alpha rarely had links, and when it did, they were not the primary content. Further, what links were displayed were selected for inclusion by people. A better Wolfram|Alpha comparison in 2009 was Wikipedia, with its summary boxes that organized its content into something more uniform for each domain.

The striking thing about Wolfram|Alpha is that this kind of information is not merely uniformly prepared canned content fed to a template. A result for, say, " $x^3 - 5x^2 + 2y^3 - 7y$ ", which the author just made up and has no particular

George Washington (politician)

Basic information:

Local historical calendar ▾

full name	George Washington
date of birth	Friday, February 11, 1732 (Julian calendar) (283 years ago)
place of birth	Westmoreland County, Virginia
date of death	Saturday, December 14, 1799 (Gregorian calendar) (age: 67 years) (215 years ago)
place of death	Mount Vernon, Virginia

Image:



Leadership positions:

official position	country	start date	end date
Senior Officer (Department of Defense)	United States	July 13, 1798	December 14, 1799
President (1 <sup>st</sup> )	United States	April 30, 1789	March 4, 1797
Continental Army General and Commander in Chief (Department of Defense)	United States	June 15, 1775	December 23, 1783

Timeline:

include today



Fig. 11.1 Partial Wolfram|Alpha result for “George Washington”

significance, is just as meaty, including a 3D plot and a contour plot (see Fig. 11.3), as well as alternate forms, roots, polynomial discriminant, integer roots, properties as a function, roots for the y variable, derivative, indefinite integral, local minimum, local maximum, definite integral over a disk of radius R, and definite integral over a square of edge length 2L, and it promises more if given more computation time (a feature available to Wolfram|Alpha Pro users).

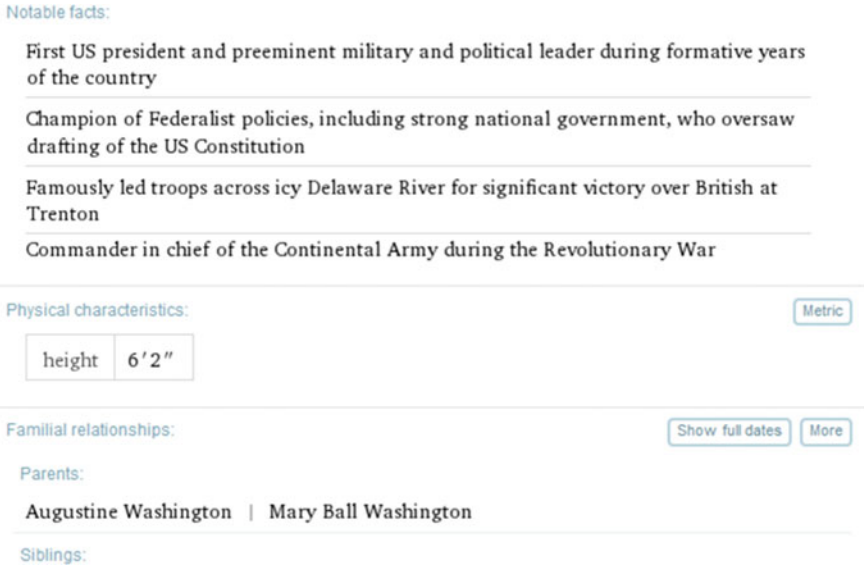


Fig. 11.1 (continued)

Even this ability to create novel content doesn't show Wolfram|Alpha in its strongest light. Its most powerful feature is that it can link facts with computation, to give a sense of proportion. Consider "unemployment rate New Mexico, Nebraska", a partial result of which is shown in Fig. 11.4. There are several striking things about this. First, you are not easily going to get a direct comparison between these two states elsewhere. The combination is novel, assembled to meet the query. The next thing that the results not only include the rate, they also offer a calculation of the change in rate over time. In an additional result segment that is not included here, there are also some absolute counts of those employed and the rank of the states in those counts. What this example shows is that novel queries to W|A that combine different entities of the same kind and receive both customized and integrated results, along with a further quantification in rates and ranks that gives those results context.

Wolfram|Alpha is also useful for discovering facts that are not explicitly present, but available through formulas. Consider the result of "two drinks over an hour for a 160 lb male", as shown in Fig. 11.5. This result shows the blood alcohol content as it relates to the legal driving limit of the United States.

Wolfram|Alpha is different from Google not only in what it does, but what it does not do. If you ask Google "who is the best president of the United States", it will return documents with historical comparisons, polls, and opinions. However,



# George Washington

1st U.S. President

George Washington was the first President of the United States, the Commander-in-Chief of the Continental Army during the American Revolutionary War, and one of the Founding Fathers of the United States. [Wikipedia](#)

**Born:** February 22, 1732, [Westmoreland County, VA](#)

**Died:** December 14, 1799, [Mount Vernon, VA](#)

**Height:** 6' 2" (1.88 m)

**Vice president:** [John Adams](#) (1789–1797)

**Presidential term:** April 30, 1789 – March 4, 1797

**Spouse:** [Martha Washington](#) (m. 1759–1799)

Fig. 11.2 Google summary of “George Washington” search

Wolfram|Alpha does not index documents or generally deliver answers with opinions or conjectures. You should expect from Wolfram|Alpha what you might expect from a very adept computer program, and there is no agreed-upon computational definition of what it means to be the best president. For this reason, although Wolfram|Alpha may have to weigh in on disputed matters of fact such as borders between countries or topics where the science has not yet been settled, it does not participate in many controversies.

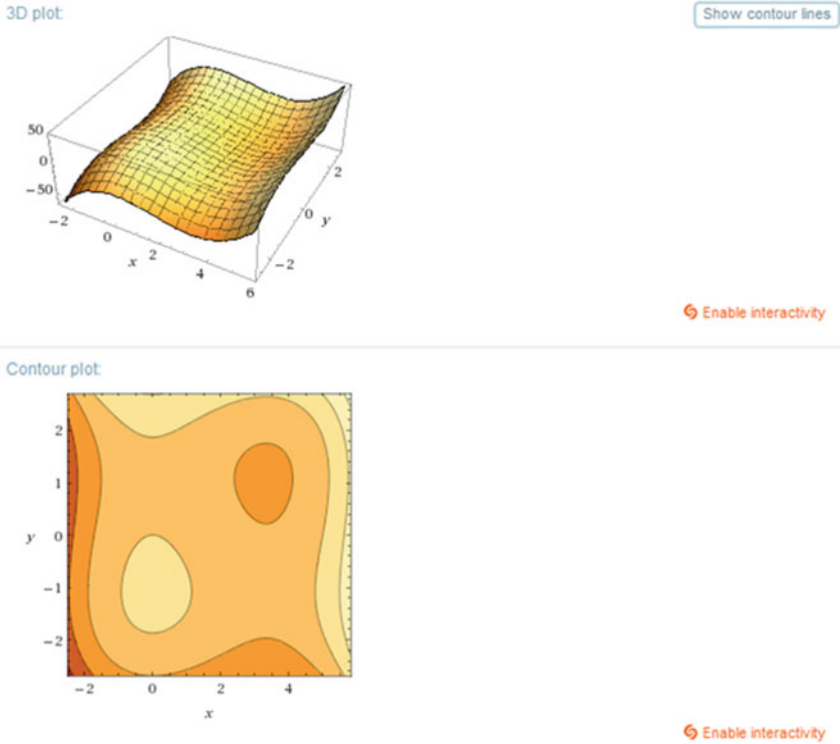


Fig. 11.3 3D plot and contour plot from “ $x^3 - 5x^2 + 2y^3 - 7y$ ” W|A query result

As Wolfram|Alpha and Google have complementary offerings, would Wolfram|Alpha + Google be a better, more comprehensive search technology? It is a matter of individual taste, but more is not necessarily better. Once one knows Wolfram|Alpha is capable of answering a question in a particular domain, one can use it to get the answer straightforwardly without sorting through other results. Similarly, there is no point in using Wolfram|Alpha when searching for articles or opinions.

Overall, Wolfram|Alpha aims not to find documents that contain facts, but to combine facts in novel ways for comparison, or even assemble facts out of formulae, while staying within the limits of what we might reasonably credit to algorithms run over facts. Throughout this text, we'll see further examples of Wolfram|Alpha making calculations to achieve novel and salient results.

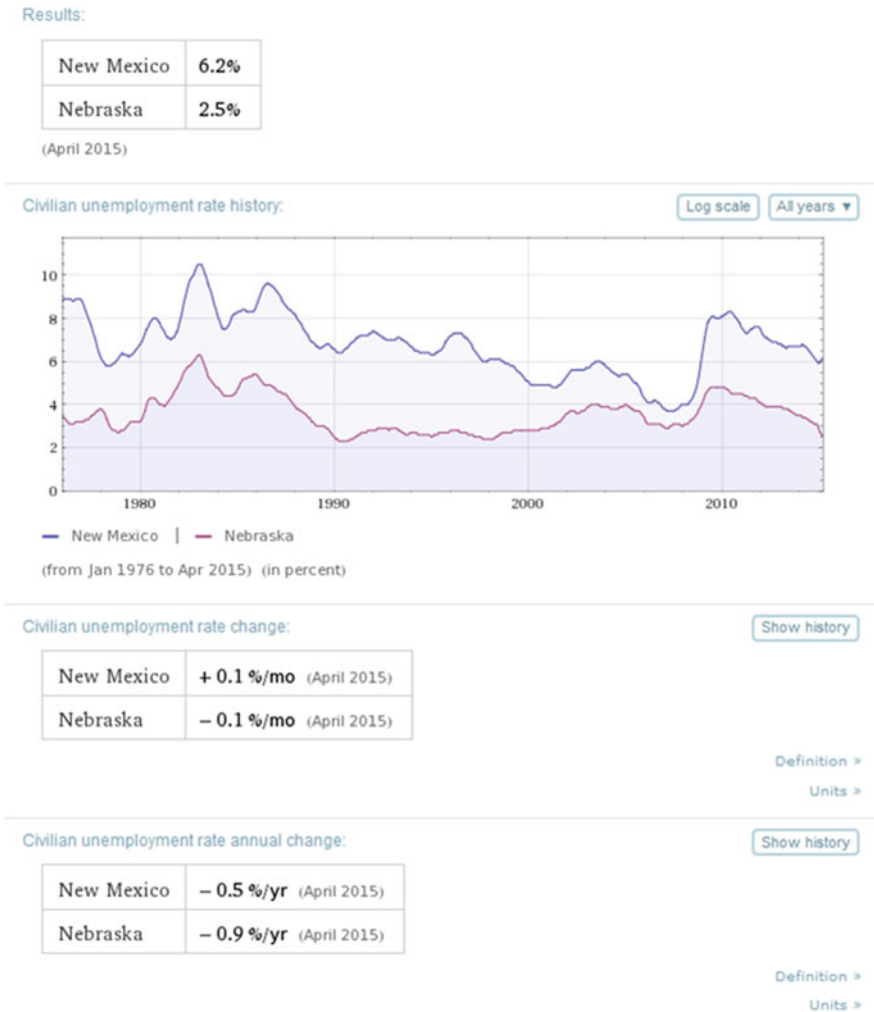


Fig. 11.4 Partial Wolfram|Alpha result for “unemployment rate New Mexico, Nebraska”

### 11.4 Internal Differences: Wolfram|Alpha in Comparison with IBM’s Watson

We might be led by these differences from Google to say that, instead of being a search engine, Wolfram|Alpha is a tool for question answering. This would be entirely fitting, as Wolfram|Alpha poses itself as a “computational answer engine”. The appropriate comparison would then be with IBM’s Watson, the system known

Input information:

number of drinks	2
gender	male
time	1 hour
body weight	160 lb (pounds)

Estimated result

blood alcohol percentage	0.05%
legal driving limit (United States)	0.08%

(estimated ignoring variations in metabolism, food intake, medication, etc. and assuming a drink contains 0.648 fl oz of alcohol)

Time plot

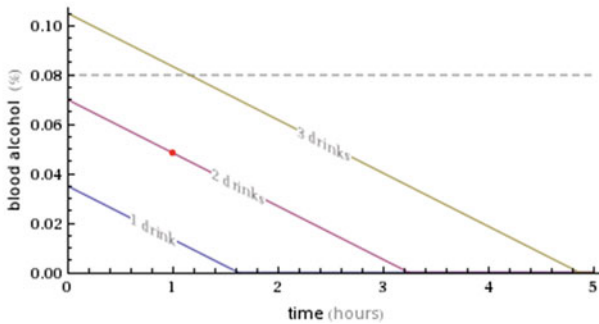


Fig. 11.5 W|A result for “two drinks over an hour for a 160 lb male” query

for being the first computer to win at Jeopardy. While those systems’ intended applications are indeed closer, there are still critical differences in their internals.

The primary difference between the two is in information retrieval versus symbolic processing. Watson uses information retrieval technology, processing queries into searches over text indices and applying statistically trained result selection. Wolfram|Alpha works rather differently. It is helpful to realize that Wolfram|Alpha was developed by the same company that wrote the Wolfram Language, a very large programming language. The initial phases of interpreting a query do not involve going to particular indexes, but is effectively a compiler for a very ambiguous programming language. The ambiguity between surviving parses is then managed by a scoring system that internal curators have tuned to yield the





Fig. 11.6 *Mathematica* 10 “single equals” evaluation of “creation year of Starry Night”

most likely answer while also offering a mechanism for the user to select from other valid interpretations of their question. These parses are then translated into a symbolic representation, in a way made usable by the public in *Mathematica* 10. For example, let us ask about the computable form of “creation date of Starry Night”, which will be understood as pertaining to the painting by Vincent van Gogh, with results shown in Fig. 11.6.

What’s happening is that this linkage to Wolfram|Alpha has converted the text to a function. It is then the processing of that function, and processing that determines information related to that function, that prepares the pages shown in Wolfram|Alpha. Later, we will explain the reasons for this methodology choice, which comes from both the technical history behind Alpha and how this allows integration into yet broader systems.

## 11.5 Institutional Differences: Wolfram|Alpha in Comparison with Wikipedia

As Wolfram|Alpha does not search external resources, which might have no clear transformation into the symbolic relations described above, it works from an internal compendium of databases and data feeds. In that way, Wolfram|Alpha is like Wikipedia, where its knowledge resources are part of the overall system. However, unlike Wikipedia, its content is not crowdsourced, save for some processed feedback from users and from some volunteer programs, but instead comes from a process of internal curation.

Wolfram|Alpha does not have open data as its implementation provides little opportunity for the rewards of open authorship, namely community recognition. Wolfram|Alpha’s data is not presented directly, but combined in multiple ways by means of computational expression. Without a narrative or composition, the pleasure of having presented one’s composition is not as preserved. Further, certain kinds of disputed data (such as contended borders between countries) need a presentation outside of a text which could nuance and qualify those disputes. Finally, numerical data, being outside of a narrative, might be too easily altered by someone with poor intentions. Overall, instead of attempting to build rewards into a

computational system that does little for authorship, Wolfram|Alpha maintains an ethos of professionalism, the comradery of community, and an aesthetics for nuance in curatorial practice from within the traditional bounds of the firm.

## 11.6 Wolfram|Alpha's Relationship with Artificial Intelligence

As we have discussed, Wolfram|Alpha's contributions to artificial intelligence are in a scalable infrastructure for syntactic question answering and the corresponding demands in knowledge representation. However, Wolfram|Alpha's differences from common artificial intelligence practices, in particular its distance from the statistical techniques dominant over the last 20 years, is perhaps more interesting.

Unquestionably, the task of answering queries posed in natural language is an artificial intelligence project. The need of Wolfram|Alpha was very different than posed in question answering tasks. Question answering is typically focused on being able to answer inquiries about a particular text, having extracted relationships. With Wolfram|Alpha, the kinds of domains and relationships that can be addressed are planned ahead of time: can Wolfram|Alpha be sufficiently comprehensive in a particular area? Does every kind of computation the W|A team chooses to make available lead to a correct answer? Among the possible interpretations of a given statement, are those that fit with W|A's potential answers plausible? The basic guideline that W|A is engineered for is delivering high precision (a minimum of false positives) at the cost of recall (a minimum of possible, but missed, answers).

This position means that nearly all potential translations to computational form are as intended, and those that are not can be remediated through clear rules. Scoring different outcomes is still necessary, but this is because some interpretations are more useful than others, even if all are semantically correct. For example, the query "bra size chart" could validly mean one wants a graph of how the population of the city of Bra in Italy changed over time. The city population area interpretation is therefore allowed, but the garment interpretation is scored higher. If interpretations are close in score, only one will be immediately displayed, but an assumptions section will be displayed that allows the other interpretations to be easily selected. Thus far it has simply been more efficient to have scoring rules for such cases than trying to introduce the floating point overhead of probability calculations into the tight loops of the parse itself.

What the Wolfram|Alpha team has done is develop efficient procedures by which an organization of people can combinatorially assemble the kinds of relationships that might be the output of natural language processing techniques, and representations appropriate to that assembly. How did the team come to this particular set of technical choices? Let us now look into the forces that made Wolfram|Alpha.

## 11.7 How Wolfram|Alpha Could Happen: A Social History

How could a technology so against the popular tide be developed? What we will see is that Wolfram|Alpha directly addressed business needs while opening up a new area of development. These business needs were able to be addressed by particular organizational processes forming in the wake of a particular event, all harnessed through existing precedents and control structures. These new processes, combined with a palette of existing and new technological capabilities and focuses, led these counter-trend technical directions to be the company’s dominant path. This development was then sustained financially with the aid of longtime strategic partners whose needs the technology supported. This chapter will take you through how the Wolfram|Alpha organization could and did make startlingly original choices.

## 11.8 The Business Context of Wolfram|Alpha

Wolfram|Alpha came along at a key time for what employees then called Wolfram Research. At that point, the conglomerate was clearly one company and had long sold one flagship product, *Mathematica*, a comprehensive platform for technical computing. Originating about 20 years before W|A, the product made a huge splash, offering new heights in symbolic mathematics, function visualization, and interactive technical document creation. In the next 20 years, growth in the company was synonymous with growth in this product, which holds a comfortable niche with competitors *Matlab* and *Maple*. It was not cheap, but you got what you payed for, backed by a dedicated group of mathematicians and computer scientists bent on integrating as much technical computing as they could into a single unified language and experience.

With such a mature product, one has to be very vigilant for competitors, particularly the ones that might initially be described as inferior. Clayton Christensen’s popular theory of disruptive innovation [1, 2] posits that many products are undermined not by superior competitors, but inferior competitors that serve customers in a different way. First, when an inferior competing product is launched, the company with the superior one does not worry too much, as very few users of the new product would buy the superior product anyway. Often, the inferior product has no real competition in its submarket. Slowly, the inferior product improves, taking progressively larger shares of the overall market, but at no time does it make sense for the superior product’s company to make a cheaper offering, as its own higher-margin users would buy the lower margin product, eating away at that company’s own profits worse than by its competitors. Eventually, the competing product can become good enough that whatever new need it was created to fill (say a program to run on mobile devices instead of desktop programs, or a disk drive smaller in physical size) will tip the overall balance of features would tip to the competitor, leaving the once-superior

product's company to be playing catch-up in a new technological substrate. Often, there is little sign that such a disruption is taking place until it has occurred, making the theory critical to recognizing potential trouble.

Mathematica was starting to show signs of being a candidate for disruption. University students often preferred scientific calculators to software programs, given their feasibility for individual classroom use and their not requiring any additional hardware except batteries. The Internet and open-source licenses meant that special-purpose university-written languages such as GAP were viable alternatives for mathematicians in particular fields. Perhaps most daunting of all, open-source general-purpose languages, such as Python, with easily extended library systems were starting to sprout numerical and symbolic computing extensions, such as numPy, that were nowhere near as powerful as *Mathematica* but were practical for some applications and continually gaining in capability. The capabilities of *Mathematica* over-served the needs of these products' users.

In the Wolfram group's business landscape, Wolfram|Alpha functions as a disruptor to potential disruptors. The Internet could now generally be assumed, and smartphone technology was progressively widespread. A web page or downloadable app that did everything the calculator could, and more, meant upstaging the calculator's market. This sort of system also meant that instead of writing their own software, people faced with complicated technical problems could enter them in plain text and receive not only an answer, but all kinds of useful related information, such as the steps by which such problems might be solved. Generally speaking, the users who would have bought Mathematica to undertake these relatively simple tasks would be few, but W|A represents direct competition for *Mathematica*'s disruptors.

The icing on the cake was that Wolfram|Alpha would not only disrupt certain kinds of competition, but could provide natural language technology for the Wolfram Language, *Mathematica*, and other products. This would allow one to write Wolfram Language programs that used natural language phrases rather than code in sections. In other words, natural language technology would form a different basis for competition that no competitors would be prepared to follow, allowing it to act as a disruptive presence in new areas.

## 11.9 An Organization that Made Wolfram|Alpha Possible

Any number of firms recognize that they have problems with disruption, but most find it nearly impossible to counter due to existing organizational commitments. R&D staff are busy creating new incremental developments that retain and grow the company's largest-margin customers. Sales staff are more motivated to pursue sales that have larger margins. How was Wolfram Research able to build Wolfram|Alpha and become the Wolfram group after 20 years of pursuing a single platform?

The true secret behind the organizational capacity to create Wolfram|Alpha was the book *A New Kind of Science*, and its accompanying research program, both

known by the abbreviation NKS. The impact of NKS was not only, or even mainly, its ideas but instead its human-resource effect on Wolfram Research. This work attracted a great number of very bright people eager to build something completely new and demonstrate the power of what they were working on. Though highly praised and hugely influential on some, the book also received neutral and negative reviews. With the book so new and not immediately adopted by skeptical and already-preoccupied professors, young physicists wanting to get involved were not immediately swallowed up by other research programs, but turned to Wolfram Research itself. It was in NKS summer schools that many key developers for W|A were discovered.

These individuals were often directed immediately into special projects or ambiguous research positions instead of fixed roles with binding job descriptions. This fluidity meant that a large part of the W|A work force was not in specific roles, but was ready to be mobilized. Further, even for *Mathematica*, it was not uncommon to organize the developing functionality of future releases through a portfolio of projects allowing a robustness to one or the other not working out. These flexibilities were critical to providing the necessary staff.

With this new influx and its corresponding projects, there was privately a fair amount of concern from longer-term company developers about this new project, with *Mathematica*-related work seeming the more stable option. However, if there was any direct budgetary competition between the company’s initiatives, it was well hidden. This centralization of fiscal discretion was critical for fostering Wolfram|Alpha’s disruptive presence. An important part of that fostering was equal access to a number of key services, including release engineering, systems administration, accounting, human resources, and other corporate components that allowed the internal start-up to begin with many institutional problems already solved.

It must be said that, even prior to the influx of NKS-influenced physicists and mathematicians, Wolfram Research staff had long had a variety of backgrounds other than computer science. Of course, given *Mathematica*’s initial framing as a tool for mathematics, mathematicians were strongly present. Physics, itself intensively requiring technical computing, has also been well represented. The base of Champaign, Illinois, home of the University of Illinois at Urbana-Champaign (UIUC) is a highly-regarded engineering program drawing its student population heavily from Chicago. As is inherent to businesses in college towns, it attracted a variety of people who came to teach and study but then moved on to something else.

As an example of the limited way in which those with conventional backgrounds played a role in W|A, a content director met a new developer in the hallway. He asked him what he had studied in school. “Computer Science” was the reply. Hmm, not physics, nuclear engineering, mathematics, or chemistry? How rare? Was he perhaps one of the few hired to work with the data frameworks? And so he was.

As another example, a key Wolfram|Alpha engineer who had been a string theorist was initially given the task of working on text-processing functions, known as string manipulation. This might sound like a bad game of telephone gone wrong, but it was entirely in tune with the organization’s strategy of turning smart people loose on comparatively simple problems.

This wave of new researchers produced a tremendous amount of technology, but it might never have been harnessed for the practical application of Wolfram|Alpha if not for another aspect of Wolfram Research, namely its history of hiring librarians. UIUC is not only highly ranked in engineering but is a top-ranking library school. The company had long had a company library and actual librarians, who would sometimes take on new roles. As the Alpha project progressively moved from its early phases to needing a definite conclusion, the company turned to an executive known for her ability to close difficult projects, who had happened to be the company's first librarian.

The gradual hiring of a department of library professionals and domain experts marked the institutional gestation of Wolfram|Alpha. These "data curators" worked in tandem with developers to shape and clean the data, and to give the phrases that would designate that particular terms of a query should evaluate to a particular term of a particular kind of function. Slowly, the group of developers improvising their tools to get things done bifurcated into "content developers," pursuing new domains and "framework developers," who added new development and curation tools, as well as new runtime features.

The establishment of a data curation department was the first step in Wolfram|Alpha settling down into a more stable organization. Content managers specializing in different kinds of knowledge brought a unified view to mathematical or socio-economic topics. Standard review processes were set up, with upper management participating only at key points in the beginning and end of the process. A systematic, frequent, and doggedly pursued release cycle provided an internal clock that assured new features were carefully engineered to be integrated quickly and that severe bugs were not allowed to linger. A team engaged with the feedback of external users, making sure they received responses and closing the loop between domain development and user needs. Log analysis provided another mode of user feedback. The project was now being managed for incremental improvement, its immediate disruption having been manifested.

This form of organization led to the conventional way that Wolfram|Alpha works [3]. Whether through logs, feedback, commercial demand, internal use, discovery of a convenient data source, or other means, a realm is identified as a viable area for potential development. A developer is paired with a librarian, one of them typically having strong knowledge of the domain. They are given relevant query logs and, with the help of the content manager, imagine potential other queries users might submit for the new content domain. Together, they research myriad pertinent kinds of facts associated with the domain. Once those are decided upon, the developer focuses on structuring the data, evaluating relevant expressions, and presenting the results while the librarian vets facts and makes all viable expressions of the domain find the correct functions. If they come up against limits or questions in expressing what is desired, they collaborate with framework developers to engineer what is needed. Depending upon the domain and the nature of available sources, the content developers and librarians can employ a great deal of automation in preparing their work, developing more automation themselves when appropriate. Therefore, advances in any technology that might help them,

such as improvements in artificial intelligence, will either first develop on a domain-specific basis, become readily provided through existing frameworks, or appear through changes in the Wolfram Language itself, and thus not seem disruptive but part of the regular workflow. The establishment of W|A’s organization is not limited to development, but also to how domains are reviewed, how current events from news or the internet (such as change in a country’s political leadership) are quickly incorporated, how the site is monitored, how quality is assured, how consulting projects are created, and any number of other matters that allow Alpha to continue to run smoothly.

The combination of NKS-driven physicists and library professionals meant that there was an absence of computer scientists or startup-interested business professionals, an unusual climate for a startup-like setting. This meant that Web 2.0’s “architectures of participation” were not an ideological requirement. Instead, the proclivities stereotypically associated with these disciplines, a personal control over correctness and organization, were actually fully manifested by the team, institutionalized rather than outsourced. This separation from technical trends also meant that Alpha’s technology was path dependent upon then Wolfram Research’s existing technology and its surrounding norms, which we will turn to now.

## 11.10 Forming the Technology that Forms Us

The dual origin of Wolfram|Alpha technology is clear: in “computable” data and in projects to support flexible calculators. By version 6, Mathematica had embedded computable data for a variety of domains. What is meant by “computable data”? Technical computing often needs access to mathematical and physical constants, and basic socio-economic uses benefit from having commonly used data and statistics available. The question was how to take on those requirements with the Wolfram Language’s “everything-included” maximalist philosophy. The answer was to allow each domain of data correspond to its own function, each with a common design to aid in documentation and overall usability. These functions could be set up to yield different results at different times, and not evaluate if given unbound variables, keeping the result of asking for data “symbolic”.

Even earlier, the need for flexible calculators was becoming clear. How could people calculate an integral without either using a typeset entry palette or remembering the order of Integrate’s arguments? If users could get a text box that corresponded to an integral, there would probably be a relatively fixed number of ways they might try to enter what they wanted to integrate, and those ways would be easy to pick out. From there, the thought was maybe it would be easy enough to tell what somebody typed in for an integral from what was typed in for a derivative, or other mathematical functions. This thought extended “Integrate” to “Calculate”, after which project name changes would be more for external than internal communication.

The idea and then pursuit of being able to include computable data in free text calculations is an example of scope creep elevated to a software-engineering

methodology. The conversion of “Integrate  $x^2$  from one to five” to `Integrate[x2, {x,1,5}]` is perhaps harder than converting “population of france” into `CountryData[“France”,“Population”]`.

Though it may seem clear that Wolfram|Alpha is, from an implementation perspective, a direct descendent of these computable-data and free-text-entry projects, why was computable data pursued in the first place? Why should a symbolic `CountryData[country,“Population”]` be considered valid source code? The ethic of allowing everything to be symbolic stems from the Wolfram Language’s roots in handling symbolic mathematics. Every result is subject to the simplification of term-rewriting and if not resolved is left unevaluated for future substitutions if at all possible. The early Wolfram Language was actually a term-rewriting system, with more traditional language constructs like lexical scope appearing later. As we’ll see later, the idea of an entirely symbolic language has been taken yet further.

With mathematics and data in place, it became clear that Wolfram|Alpha would need other capabilities, including data on formulas and how to compute them, along with units such as length and time. An early uncanny result of Wolfram|Alpha’s integration between data, units, and mathematics occurred when a software quality engineer tried the query “cubic lightyear of jello” and it worked! (See Fig. 11.7).

How could this complex interaction between computable data and free text work, given the variety of different processing needs for different kind of data? How was Wolfram|Alpha able to take advantage of domain-expert developers without much

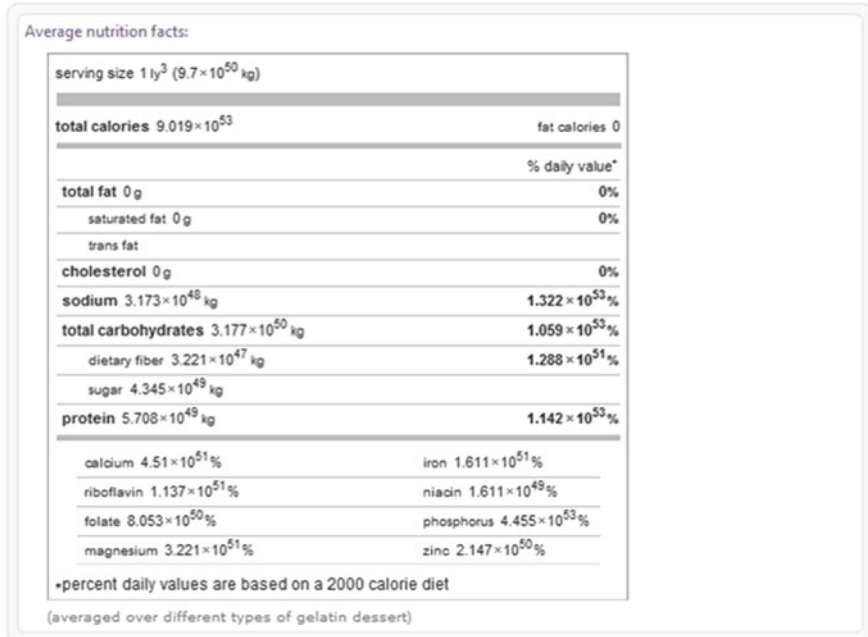


Fig. 11.7 Wolfram|Alpha “cubic lightyear of jello query rendered in *Mathematica*



particular training in databases or information retrieval? The secret was to extend the designed commonality across various data functions like CountryData and ElementData all the way down into a database of common modular parts, easily translated from Wolfram Language statements. Like tactics in a videogame based on invading territory, W|A’s data infrastructure was built in simple units that could be assembled in giant swarms to speed the initial rush of product release. It was only after this technology was deployed that the team pursued more refined platforms with greater expressiveness but more specialized uses. To this day, the simple, modular system survives, with its integration into standard toolkits and its terminology forming the dominant metaphors for developers.

These structures also created a standard metaphor that colored the kind of interaction the organization took on with unstructured data. Wolfram|Alpha didn’t need to automatically understand messy data. The question for the team was always this: Can we get the data into forms we want to work with relatively simple tools? Dealing with unstructured data was a problem handled by strategic assessment rather than raw processing ability.

### 11.11 Partnerships: “Help from Our Friends”

Though Wolfram|Alpha’s role in Wolfram’s competitive landscape was clear, what wasn’t exactly clear was how it was going to make money. Of course, there was advertising on the website itself. The company offers a paid Wolfram|Alpha Pro version that can analyze the users’ own data, as well as calculate step-by-step breakdowns of mathematics and chemistry problems, and process user images, among other features continually being added. Another approach has been to create appliances that allow users to query their own data with natural language, and Wolfram|Alpha servers for institutions not wishing to enter their queries over the public Internet. An API lets users to put W|A to their own uses. Widgets permit content producers to embed purpose-specific information and visualization in their contents.

All of these have contributed to Wolfram|Alpha, but partnerships have been the dominant means of support. Unquestionably, the use of W|A within Siri has been W|A’s broadest exposure. The relationship between Wolfram companies and Apple goes back to Steve Jobs having suggested the name *Mathematica*. And in Wolfram|Alpha’s early years, Microsoft played a crucial role in Wolfram|Alpha, at times integrating W|A with Bing. Overall, these partnerships enabled Wolfram|Alpha transitioning from an emerging to an established technology, through which we see the cycle from strategic need and organizational mobilization to a mature platform completed.

Now that we have seen the transition to a contemporary Wolfram|Alpha, let us look into today’s W|A itself and see some of what it does and is for.

## 11.12 What Wolfram|Alpha Does, by Example

When should you use Wolfram|Alpha? One answer is when you don't know if the content is out there, and you need facts based on the content you have. Another is when you have the facts, but need a summary. Yet another is to get a sense of proportion or scale. W|A is also useful when you want a clean view of just the information. It can also be fun to try the service just to see what it will do. Let us illustrate these points by looking into a few of the many queries Wolfram|Alpha could address. This is necessarily only a small sample; you might prefer to visit the gallery of examples at <http://www.wolframalpha.com/examples/>.

### 11.13 Example: Food

Food is one area that combines facts with mathematics. Oftentimes, people want to understand what they are eating without having to be too particular or do the math themselves. If you just happen to want, say, a hamburger and fries at some random cafeteria, W|A can create an approximate nutrition label (see Fig. 11.8).

People often want to go the other way, too, where they have some nutritional goal and want to find the foods that help get there. For instance, a person with an iron deficiency might want to know which foods have the most iron (see Fig. 11.9).

That's great, but what does this mean? We can also ask much iron a person needs per day (see Fig. 11.10).

### 11.14 Example: Automobiles

Wolfram|Alpha can also offer capable summaries. Automobiles are a good example for this. Let's start with the most basic case, a single car, let's say the "Ford C-Max Hybrid". Wolfram|Alpha provides an image and a summary (see Fig. 11.11) as well as breakdowns for price, fuel efficiency, engine and transmission information, interior and exterior dimensions and available colors, safety ratings, warranty provisions, awards, features, available rebates, and Wikipedia hit history.

An overview of a particular car can be quite useful, but often we are interested in comparing the merits of one car against another. Let's compare "Ford C-Max Hybrid versus Toyota Prius", this time excerpting (Fig. 11.12) from down the page where we can see price and fuel-efficiency comparisons. We can see that the C-Max is modestly more affordable while the Prius has superior fuel economy.

In addition to model-by-model comparisons, we can compare automotive divisions as a whole, such as "Ford versus Toyota", to really get a sense of the differences between the two brands. In addition to the body style, price, and

Input interpretation:

hamburger	amount	1 hamburger	+
french fries	amount	1 serving	

Total nutrition facts:

Individual nutrition facts

serving sizes (total: 295 g)	
hamburger: 1 hamburger (161 g)	
french fries: 1 serving (134 g)	
<hr/>	
<b>total calories</b> 740	fat calories 305
<hr/>	
% daily value*	
<b>total fat</b> 34 g	<b>52%</b>
saturated fat 10 g	<b>50%</b>
trans fat 1 g	
<b>cholesterol</b> 57 mg	<b>19%</b>
<b>sodium</b> 1 g	<b>43%</b>
<b>total carbohydrates</b> 82 g	<b>27%</b>
dietary fiber 34 g	<b>136%</b>
sugar 8 g	
<b>protein</b> 26 g	<b>52%</b>
<hr/>	
vitamin A 3%	vitamin C 25%
calcium 17%	iron 28%
vitamin D 6%	vitamin E 5%
thiamin 36%	riboflavin 22%
niacin 50%	vitamin B6 26%
vitamin B12 33%	folate 26%
phosphorus 32%	magnesium 18%
zinc 31%	
*percent daily values are based on a 2000 calorie diet	

Fig. 11.8 Partial Wolfram|Alpha result for “hamburger and fries”

fuel-economy comparisons shown in Fig. 11.13, W|A also compares the engine volumes and physical dimensions of the companies’ vehicles.

### 11.15 Example: Sports and Games

A different area with a lot of possibilities for quantitative comparison is sports and games. We might be interested in extreme events, such as “2014 MLB game with most hits”, excerpted below (see Fig. 11.14).

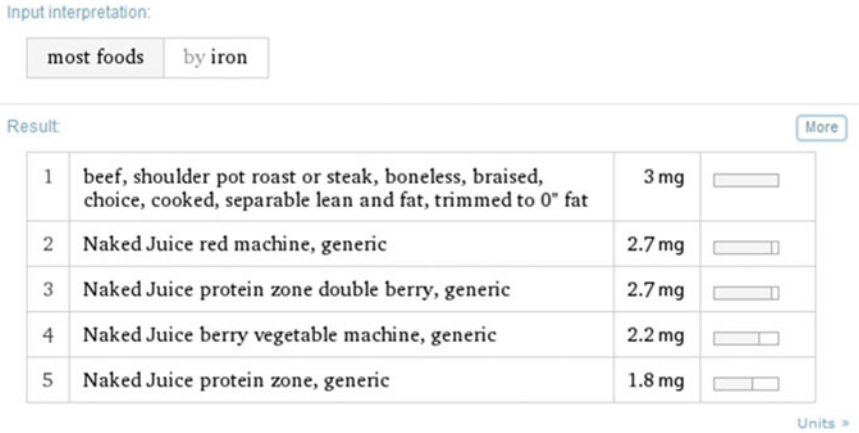


Fig. 11.9 Wolfram|Alpha result for “foods with most iron”

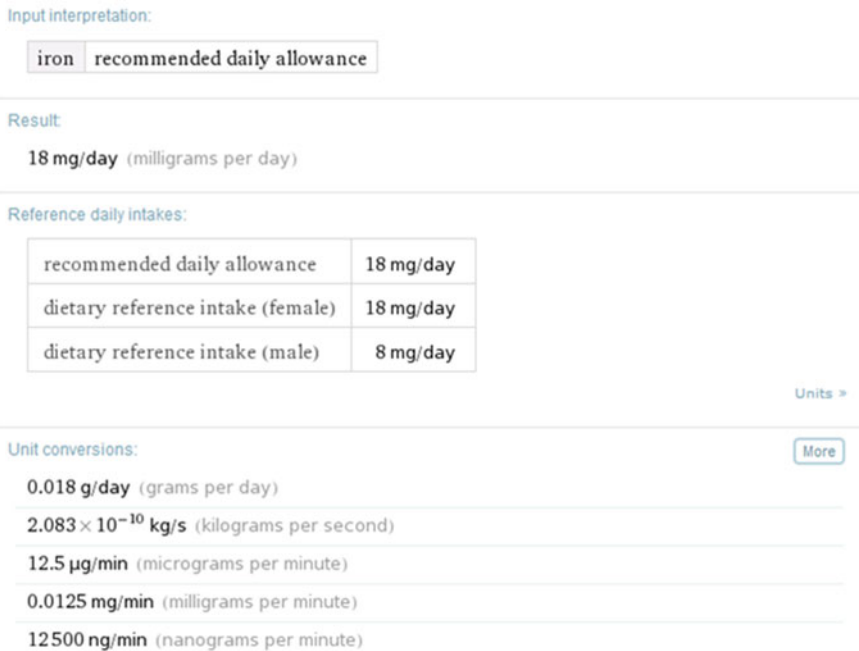


Fig. 11.10 Wolfram|Alpha result for “daily recommended iron”

We might also be interested in, say, understanding the total scale or variety in a particular game. Consider “(Most powerful pokemon hit points)/(least powerful pokemon hit points)”, which offers one window into how balanced the various

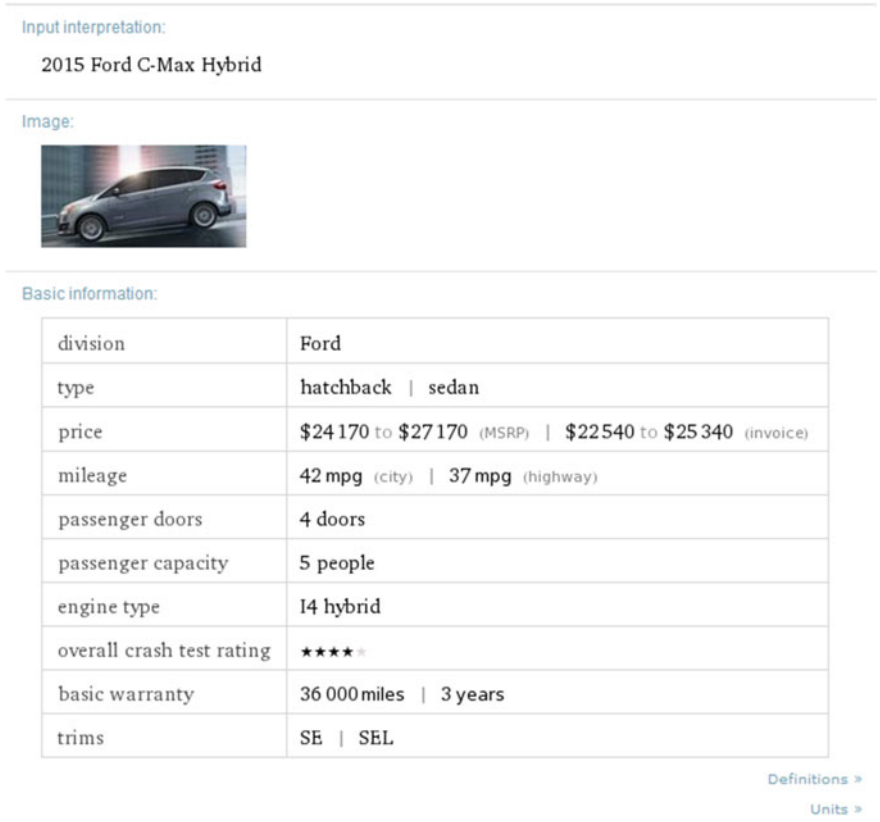


Fig. 11.11 Partial Wolfram|Alpha result for “Ford C-Max Hybrid”

characters in the Pokémon game are. This result is very direct, with Fig. 11.15 showing the complete output for this query.

Wolfram|Alpha’s information on games is not just confined to understanding their attributes—it can occasionally offer some direct assistance. Consider how “\_al\_la\_” is interpreted as a request for those English words that can fill the implied blanks of the underscore, for help with crossword puzzles, hangman, and the like. Figure 11.16 shows the complete result.

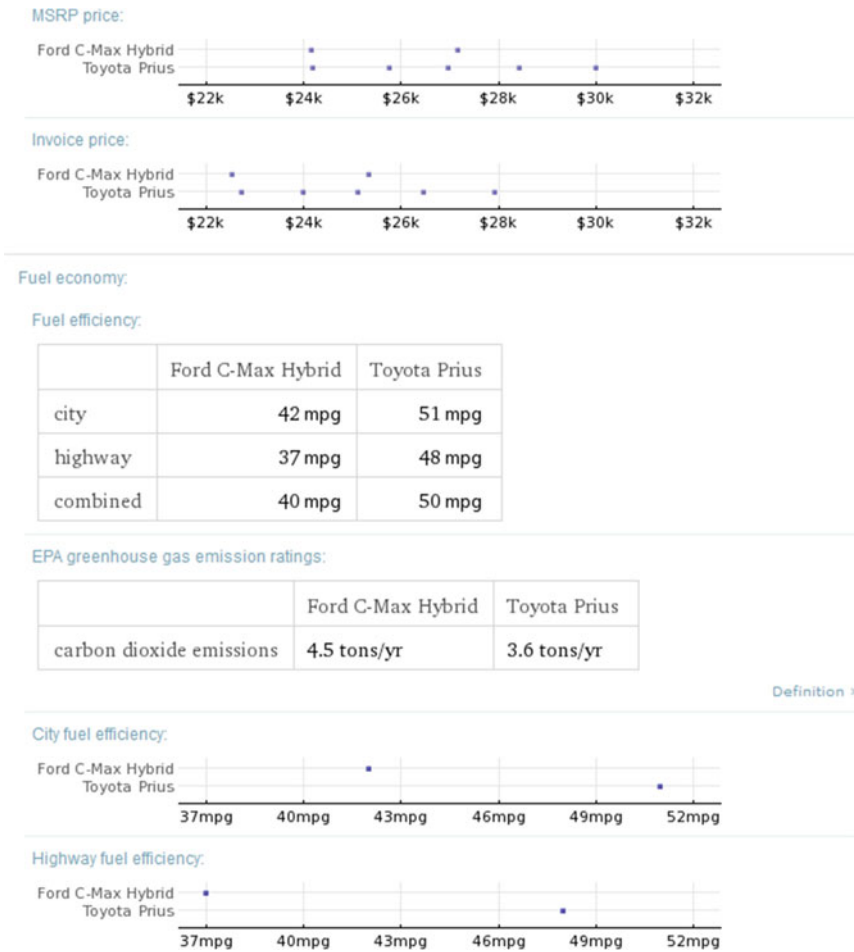


Fig. 11.12 Price and fuel-efficiency comparisons from “Ford C-Max Hybrid versus Toyota Prius” Wolfram|Alpha query

### 11.16 Example: Places

In addition to offering quantitative comparisons between things and information on events, Wolfram|Alpha also offers geographic, economic, social, and meteorological insight into places (and not just on Earth). A friend of mine is now living in Santa Rosa in California, so I decided to compare Santa Rosa with Champaign, Illinois. “Champaign, Santa Rosa” returns some basic facts about the two cities, including the segment shown in Fig. 11.17.

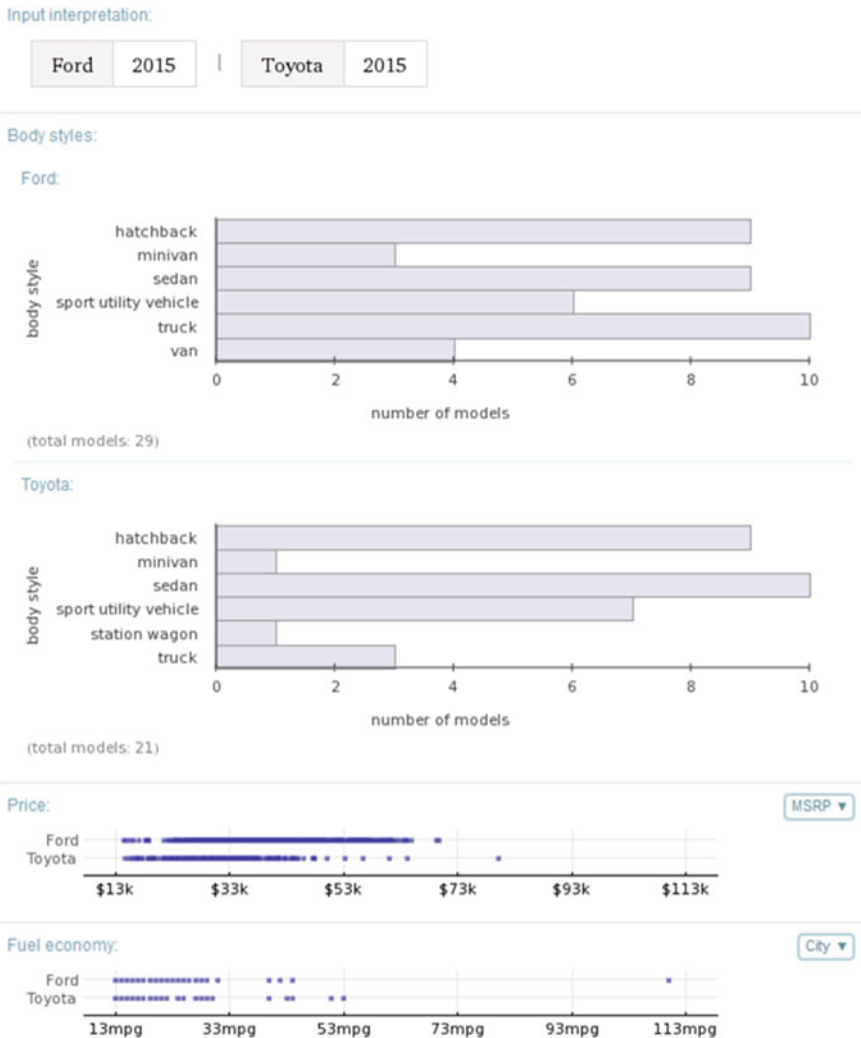


Fig. 11.13 Partial result of “Ford versus Toyota” Wolfram|Alpha query

You might suspect the cost of living would be sharply higher in Santa Rosa, and “Champaign versus Santa Rosa cost of living” reveals that is true, as shown in Fig. 11.18.

Among the facts returned by a “Champaign to Santa Rosa relocation” query, we see (Fig. 11.19) the difference in some commonly incurred costs.

However, that same query also reveals that Champaign’s crime rates and sales tax are higher (see Fig. 11.20).

Input interpretation:

MLB game	with	2014 season
	by	most combined hits

[Definition >](#)

Result:

**Detroit Tigers** (63–52) at **Toronto Blue Jays** (63–56) (August 10, 2014)  
(39)

Box score:

	1	2	3	4	5	6	7	8	9	10	11
Det	3	0	1	1	0	0	0	0	0	0	0
Tor	0	0	0	0	0	2	2	0	1	0	0

	12	13	14	15	16	17	18	19	R	H	E
Det	0	0	0	0	0	0	0	0	5	22	1
Tor	0	0	0	0	0	0	0	1	6	17	2

Fig. 11.14 Partial Wolfram|Alpha result for “2014 MLB game with most hits”

Input interpretation:

most powerful Pokémon	by total stats	hit points
least most powerful Pokémon	by total stats	hit points

Results:

[Hide details](#)

4

Arceus	hit points	120
Sunkern	hit points	30

Fig. 11.15 Wolfram|Alpha result for “(Most powerful pokemon hit points)/(least powerful pokemon hit points)”

### 11.17 Example: Relations

The kinds of practical relations that can be discovered computationally are not restricted to numerical ones. Consider the result of “grandfather’s sister’s daughter”, which finds the name of the relation considered by this path (see Fig. 11.21).



**Fig. 11.16** Wolfram|Alpha result for “\_al\_la\_”

Input

\_ a l \_ \_ l a \_ \_

---

English words:

b a l a c l a v a
c a l c u l a r y
c a l c u l a t e
f a l c u l a t e
m a l t o l a t e
m a l v a l a t e
s a l e s l a d y

(7 words)

## 11.18 Example: Humor

A completely different reason for using Wolfram|Alpha is for the fun of seeing how it will respond. Any technology that answers general questions will be asked for cultural references, such as “What is the answer to the question of life, the universe, and everything?” or “What is the airspeed velocity of an unladen swallow?” Once W|A successfully responds to such questions, there is naturally an interest to see whether it will play along with requests to “Open the pod bay doors” and the like, but perhaps also to see how it will answer personal questions. For the system to play along with cultural references is shows that there are people willing to have fun behind the curtain, and do so with a certain spirit. The query “tell me a joke” usually returns a mathematical one, such as the one shown in Fig. 11.22. Notice the parenthetical “my favorite sub-genre” that attributes a personality to W|A.

There are many more possibilities. The example page lists Mathematics, Words & Linguistics, Units & Measures, Step-by-step Solutions, Statistics & Data Analysis, People & History, Dates & Times, Data Input, Chemistry, Culture & Media, Money & Finance, Image Analysis, Physics, Art & Design, Socioeconomic Data, File Upload, Astronomy, Music, Health & Medicine, CDF Interactivity, Engineering, Places & Geography, Food & Nutrition, Education, Materials, Earth Science, Shopping, Organizations, Life Sciences, Weather & Meteorology, The Technological World, Sports & Games, Computational Sciences, Transportation, Web & Computer Systems, and Surprises as further categories to explore. Visit

Populations:

	Champaign	Santa Rosa
city population	<b>84 513 people</b> (country rank: $\approx$ 403 <sup>rd</sup> ) (2014)	<b>174 170 people</b> (country rank: $\approx$ 145 <sup>th</sup> ) (2014)
urban area population	<b>123 938 people</b> (Champaign– Urbana urban area) (country rank: 215 <sup>th</sup> ) (2000)	<b>285 408 people</b> (Santa Rosa (CA) urban area) (country rank: 111 <sup>th</sup> ) (2000)
metro area population	<b>232 336 people</b> (Champaign– Urbana metro area) (country rank: 192 <sup>nd</sup> ) (2011)	<b>488 116 people</b> (Santa Rosa–Petaluma metro area) (country rank: 104 <sup>th</sup> ) (2011)

Path:

Show coordinates



Distance:

Show metric

	distance	flight time
Champaign to Santa Rosa	1838 miles	3 hours 20 minutes

(assuming direct flight path at 550 mph)

Fig. 11.17 Partial Wolfram|Alpha result for “Champaign, Santa Rosa”

<http://www.wolframalpha.com/examples> to see the ever-increasing diversity that we can only begin to sample in this chapter.

## 11.19 What Wolfram|Alpha Cannot yet Do

Yet despite all that might be within Wolfram|Alpha’s conventional ability to do, it pragmatically does only what people frequently ask for, and might be expanded at any time. “All former living Presidents of the United States” is a query that does not work at the time of this writing, but W|A has all the development and data capabilities to make it do so if it were sufficiently desired. “2 drinks in one hour for an average US weight male” is another such query that Wolfram|Alpha could be made

Results:

Champaign	98.7 (Q3 2014)
Santa Rosa	137.9 (Q2 1997)

(100 = US national average)

Relative values:

	visual	ratios		comparisons
Santa Rosa		1.397	1	39.72% larger
Champaign		1	0.7157	28.43% smaller

Cost of living indices:

	Champaign	Santa Rosa
grocery (13.48% of total)	100.5	117.5
healthcare (4.89% of total)	117.8	135.9
housing (26.05% of total)	89.6	196.6
miscellaneous (33.01% of total)	98.2	111.9
transportation (12.63% of total)	96.6	122.4
utilities (9.95% of total)	114.7	99.3

(Q3 2014; 100 = US national average)

Fig. 11.18 W|A result for W|A “Champaign versus Santa Rosa cost of living”

to solve, though notice that the input has the subtle requirement to determine the average weight excluding those below the legal US drinking age. The available content is driven by user demand as well as the availability of the data. Automobile data is a relatively recent addition, and no doubt if a sizable number of boat enthusiasts tried to use Wolfram|Alpha to compare boats, such vehicles would also be added.

Let us now talk about some functionality that is outside of Wolfram|Alpha’s scope. The input length of queries to Alpha is deliberately limited. It does not currently include the ability to enter text and answer semantic questions revealed

Average retail prices:

Show prices More

ground beef (1 lb)	1.39 × Champaign
whole milk (0.5 gal)	1.03 × Champaign
white bread (24 oz)	1.39 × Champaign
apartment rent (2 BR)	1.33 × Champaign
mortgage rate (30 yr fixed)	1.02 × Champaign
total energy	0.883 × Champaign
doctor visit (routine)	1.08 × Champaign
gasoline (1 gal + tax)	1.15 × Champaign

(Q2 1997)

Fig. 11.19 Average retail prices from W|A “Champaign to Santa Rosa relocation” result

Crime rates:

Show details Show chart

property crime total	60% less in Santa Rosa
violent crime total	36% less in Santa Rosa
total rate of crime	58% less in Santa Rosa

(UCR Part I offenses)

(1992 and 2009 estimates)

Sales taxes:

Show breakdown

	Champaign, Illinois	Santa Rosa, California
total sales tax rate	9%	8.75%

Fig. 11.20 Crime rate and sales tax information from W|A “Champaign to Santa Rosa relocation” result

from it. Nor does Alpha support ongoing interactivity, such as questions involving pronouns or other indexical references to previous queries or responses.

Right now, Wolfram|Alpha does not automatically extract facts from unstructured data, such as documents. Even simple structures, like tables, present imponderable difficulties. For one thing, documents can refer to any time, while Wolfram|Alpha queries are evaluated at the present time. Consider a document titled “Guest List” that has the entry “All of the living former Presidents of the United States”. W|A might be able to determine the current list of people, but what if this guest list is historical, or refers to a future event? It would not necessarily be appropriate to resolve this entry’s phrase to a particular present-day list of people. As another example, consider a table of material properties with an entry of “all

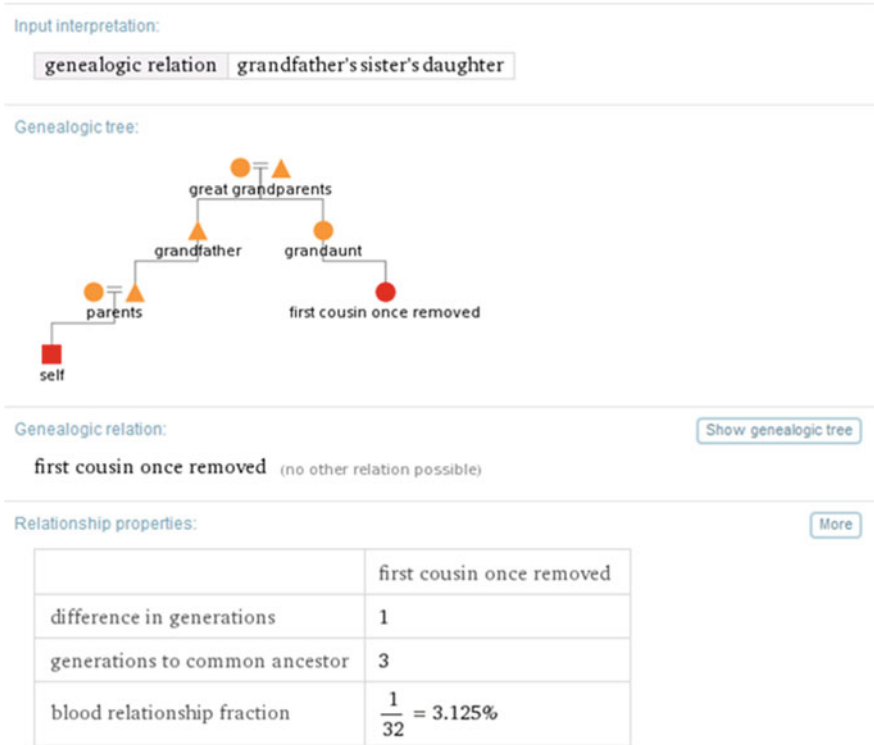


Fig. 11.21 Wolfram|Alpha result for “grandfather’s sister’s daughter”



Fig. 11.22 One of the W|A results for “tell me a joke”

plastics not otherwise mentioned”. What this phrase may refer to depends on a lot of domain expertise likely not available to any general-purpose knowledge extraction method.

As we will see later, the right way to tackle these issues might not always be further development for Wolfram|Alpha, but further enhancing Wolfram|Alpha so it can participate as a component in other projects.

In addition to what Wolfram|Alpha happens not to do and that which is extremely difficult for Wolfram|Alpha to ever do, there is what Wolfram|Alpha is not

allowed to do. One of the few cases in which Wolfram|Alpha has experienced a decline in functionality is in generating personal reports from Facebook. Here is a paragraph from a blog announcing this decline: “You’ll still be able to generate an analysis of most of your own activity on Facebook, but you won’t have access to any information about your friends (except their names) unless they’ve also authorized our Facebook app. So in most cases, we won’t have enough data to generate a meaningful friend network graph, or to compute statistics about location, age, marital status, or other personal characteristics of your group of Facebook friends.” Facebook serves as an example that Wolfram|Alpha can only undertake domains that people allow to be gathered in a systematic way.

Now that we have looked at what Wolfram|Alpha does and does not do, let us look to the new directions that Wolfram|Alpha has led to and in which it is serving fascinating new roles.

## 11.20 The Ongoing Legacy

As we have seen, Wolfram|Alpha has changed the search-engine landscape, making summaries and answers an increasingly important part of search results. Along with its impact on the general public, the existence of Alpha has changed the direction of the Wolfram group. This section looks at how W|A’s internal legacy is developing, not resting on feedback driven incremental improvement alone, but also creating new modes of interaction that challenge Alpha’s technical development in categorical ways.

## 11.21 New Initiatives

Overall, W|A Pro led to an experience of finding a new market for Wolfram Language services, which has led to Wolfram building new cloud, web, and mobile offerings aimed at different software needs with different means of collecting revenue. The idea of developers not wanting to the hosting responsibilities of webMathematica to deliver their applications has led to the Wolfram Programming Cloud, offering not just “pay for full-functioning desktop use” but also “pay as you go for application resource usage”. The Wolfram Data Science Platform will be aimed at supporting data analysis workflows. Many other such products are in the pipeline, each aimed at serving a distinct customer profile.

The experience of finding a new market has led to other changes. Wolfram|Alpha, Wolfram Finance Platform, and Wolfram *SystemModeler* have brought new customer bases which give the company a new chance to tell its story. These new audiences have led to *Mathematica*’s built-in language, itself initially called *Mathematica*, being revised and rebranded as the Wolfram Language.

## 11.22 A Language with Reference to the World

However, the most dramatic change to the Wolfram Language as a result of Wolfram|Alpha happened around *Mathematica* 10: the idea of embedding named references to entities in the world in a programming language. Consider the ability in *Mathematica* to ask for the Wolfram Language expression that returns entities for all the planets, as shown in Fig. 11.23.

What this means is that the representation of a planet is a first-class citizen of the language, in the same degree as  $1/5$  being represented by `Rational[1,5]`. For a language audited carefully for conceptual consistency, what this amounts to is a Kripke-esce declaration of the necessity of names that puts Saturn (or France, or any number of other entities) on par with `True` and `5`. From a practical perspective, this expansion means that users can do analysis across a domain, for example not just obtaining the release date and box office for a particular movie, but for all movies. These new requirements require new modes of representation and new infrastructure to support them, modes that cannot help but be suggestive of new Alpha functionality.

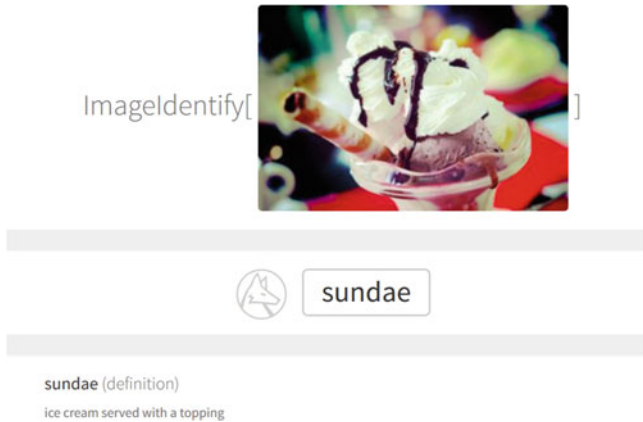
## 11.23 Finding the Semantic Meaning of Unstructured Data

Now that the Wolfram Language has semantic elements in direct reference to things in the world, it has a working metaphor for dealing with unstructured data. New Wolfram Language functions such as `SemanticInterpretation`, `Interpreter`, and `SemanticImport` all work within different internal constraints and usage scenarios for a common end: to produce these semantic annotations in an appropriately relational form, such as the new `Dataset`, from user data. Of course, users will want more than what Alpha has curated, and can provide their own symbolic item tokens as desired. This may be the future of Wolfram|Alpha development: to exploit the same means of markup available to particular markets of users.

Perhaps the most impressive extension of Wolfram|Alpha’s technologies thus far to assist other projects in interpreting unstructured data has been addition of conceptual entities for the Wolfram Image Identification Project. For example, consider the task shown in Fig. 11.24 of identifying a picture of a sundae. W|A already had to know about sundaes for the purpose of giving nutritional information, but now its

```
In[2]:= EntityList["Planet"]
Out[2]= {Mercury, Venus, Earth, Mars, Jupiter, Saturn, Uranus, Neptune}
```

Fig. 11.23 *Mathematica* 10 listing of planet entities



**Fig. 11.24** Wolfram Image Identification Project result

internal representation has been expanded to represent the “sundae” apart from its contexts as a food with nutritional content and as a word with a definition. This new computability is having explicit entities not only for proper things in the world, as demonstrated above, but for generic things. What we have seen by now is that this is but one more step in a long project of systematizing symbolic analogs to real-world phenomena in a continued fusion of librarianship with computation.

With this view into Wolfram|Alpha’s continuing future, let us now sum up what we have covered in this chapter.

## 11.24 Summing up

We have now seen how Wolfram|Alpha is different from other search and answer technologies: it does not search through documents, but instead attempting to make ordered presentation of facts related to the function the query best seems to represent. These facts are organized by professionals in an attempt to attain similar saliency to that of a knowledgeable librarian. These differences aligned with a developing need in the Wolfram group’s business strategy and were initially fostered by the combination of newly minted NKS-inspired employees and a tradition of librarianship, directed by existing technical choices, and supported by long-standing professional relationships. The queries in Wolfram|Alpha give factual answers to free text questions, providing context, proportion, and scale through comparisons and extremes, while still allowing for a sense of fun. The work of Wolfram|Alpha is not over, but is being extended to provide symbolic links to the world in tandem with new ventures.

In conclusion, Wolfram|Alpha is a people-in-the-loop organizational technology for the symbolic computation of worldly facts, developing both incrementally from



the in-product demands as manifested through logs and sharply through the demands for new kinds of applications, aimed at providing specific answers to queries across a wide range of curatable domains. Hopefully you now have an understanding Wolfram|Alpha’s niche in search and answer technologies and of its developing role in an ever-broadening range of approaches for bringing symbolic computation into the world.

## References

**To understand the role of disruptive innovation plays in business competition, consider reading Clayton Christensen’s original books on the concept:**

1. Christensen, C. M. (1997). *The Innovator’s Dilemma: When New Technologies Cause Great Firms to Fail*. Harvard Business School Press, Boston, MA.
2. Christensen, C. M. and Raynor, M. E. (2003). *The Innovator’s Solution: Creating and Sustaining Successful Growth Businesses*. Harvard Business School Press, Boston, MA.

**To understand the sociological forces that organization formation imposes on information technology practice, consider reading this classic paper by Wanda Orlikowsky:**

3. Orlikowski, W. (1992). The duality of technology: Rethinking the concept of structure in organizations. *Organization Science*, 3(3):398–427.

## Author Biography

**John B. Cassel** worked with Wolfram|Alpha from its development, through its initial release, into its stabilization, and into the new horizons of integration. He maintains interests in real-time discovery, planning, and knowledge representation problems in natural systems management, risk governance, and engineering design. John holds a Master of Design Degree in Strategic Foresight and Innovation from OCADU, where he developed a novel research methodology for the risk governance of emerging technologies.