True Knowledge: Open-Domain Question Answering Using Structured Knowledge and Inference

William Tunstall-Pedoe

■ *This article gives a detailed description of True Knowledge: a commercial, open-domain question-answering platform. The system combines a large and growing structured knowledge base of commonsense, factual, and lexical knowledge; a natural language translation system that turns user questions into internal language-independent queries; and an inference system that can answer those queries using both directly represented and inferred knowledge. The system is live and answers millions of questions per month asked by Internet users.*

True Knowledge is an open-domain question-answering platform. Behind the platform is a large and growing knowledge base of the world's knowledge in structured form combining commonsense, factual, and lexical knowledge. Natural language questions are answered by first translating the question into a language-independent query and then executing the query using both knowledge in the knowledge base and additional knowledge generated by a general inference system. The user experience is thus a direct answer to naturally phrased questions on any subject (see figure 1).

The motivation for the project was to tackle what might be regarded as the "holy grail" of Internet search, replacing larger and larger numbers of keyword-based lists of links with perfect, direct answers to naturally phrased queries on any subject. The platform was also designed to scale, with the primary mechanism for answering more and more questions being the addition of knowledge to the platform rather than writing more program code. Additional knowledge areas are typically included by adding "knowledge about knowledge."

The system is live and answers millions of questions per month, asked by real Internet users. Questions can be tried at (and API access obtained from) www.trueknowledge.com.

 ISSN 0738-4602

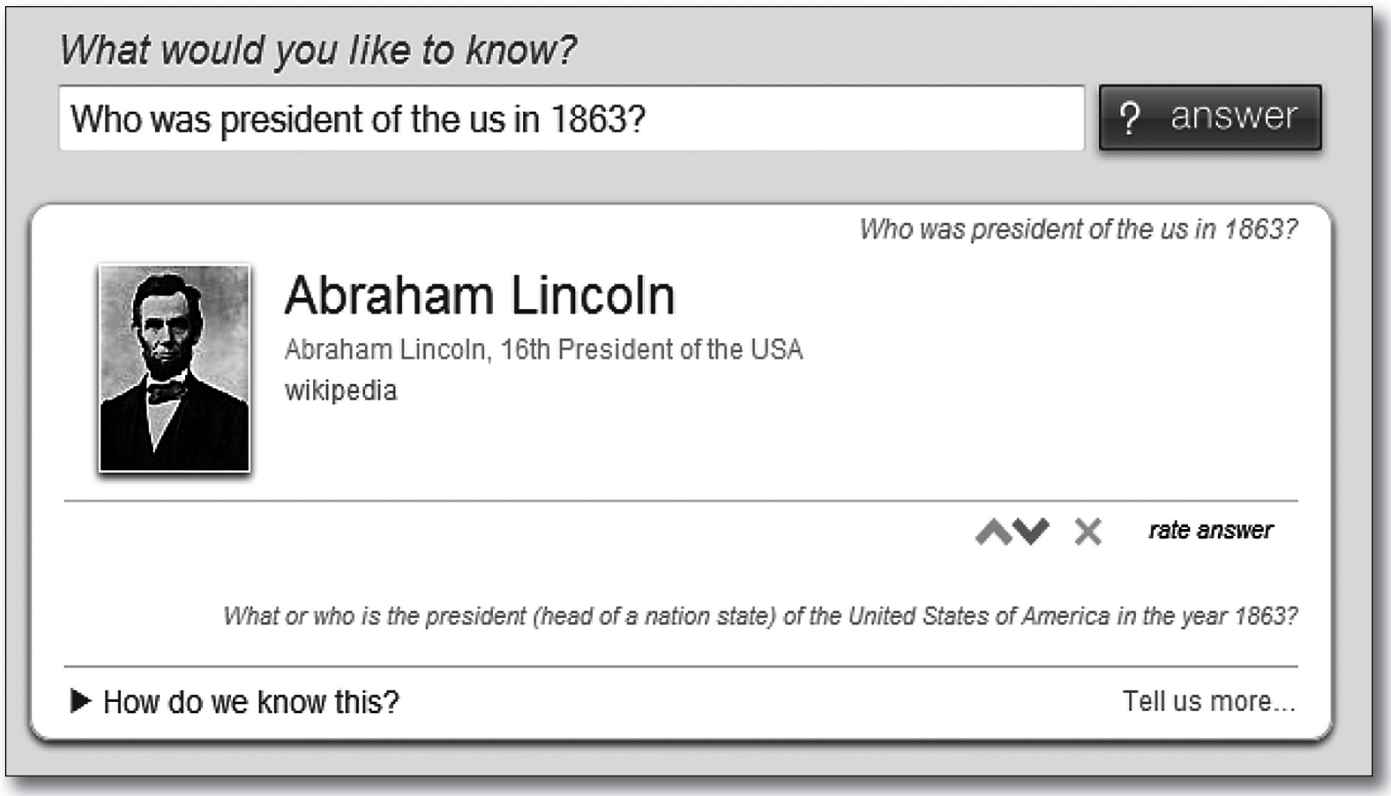

Figure 1. Example Question Response.

Origins and Current Status of the Project

The system was originally a personal project by the author, begun in the late 1990s. All the intellectual property was subsequently transferred in 2006 to a Cambridge, UK-based, startup business, True Knowledge Ltd, which subsequently received venture finance and now employs 30 people. The company's mission is to continue to develop and improve the technology, grow the knowledge base, and apply the technology commercially.

How the System Works

An overview of the architecture is shown in figure 2. Users can interact with the system using a browser interface. User questions are translated into an internal query language using the natural language translation system. The resulting queries are then executed to produce answers using both static knowledge stored in the knowledge base and facts generated by inference. External feeds of knowledge such as financial information can be brought into the platform through this system too.

Knowledge can be added to the platform by users using the browser interface too. Knowledge integrity is managed by two systems. First, user assessment allows users to contradict or endorse existing facts, optionally providing additional sources for the knowledge in the platform. Second, system assessment uses the inference system to switch off knowledge that is in semantic conflict with other knowledge in the knowledge base.

External computer systems can connect to the platform at two points through an API. Natural language can be processed directly — essentially an English question is sent to the platform and an answer returned. Alternatively, external computer systems can send True Knowledge queries directly to the platform. An example application for this second level of integration would be to obtain the local time for a given place.

All these components are described in more detail later on.

Knowledge Representation

The knowledge in the knowledge base is represented in a single unified format: named relations between pairs of named entities referred to as "facts." Facts and the relations themselves are first-class entities so facts about facts and facts about the properties of relations are fully supported. Negation is also fully supported: facts can state that a relation does not apply between two entities. The entity representation supports other entities being referenced within it, allowing various kinds of group objects and members of infinite classes to be supported. A complete temporal model is also provided by allowing facts to reference

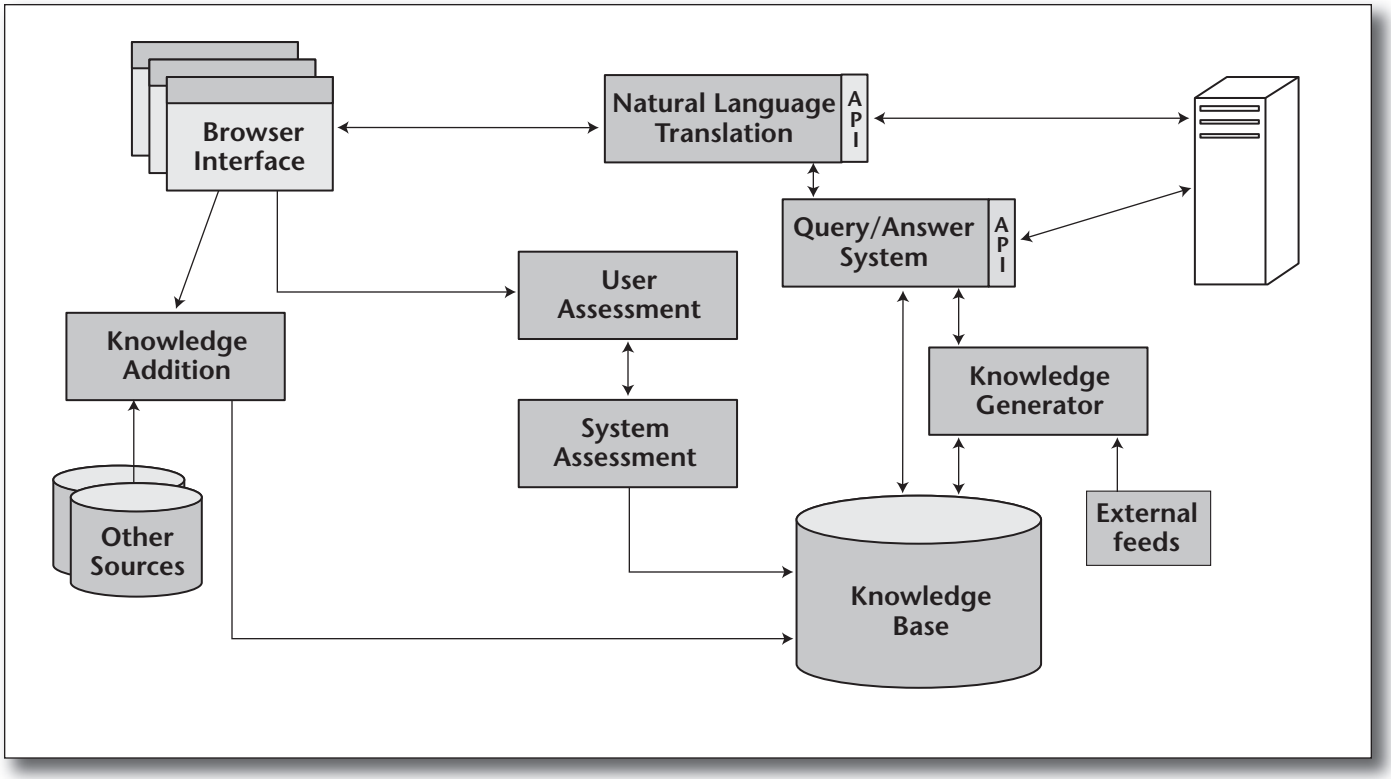

Figure 2. Architecture.

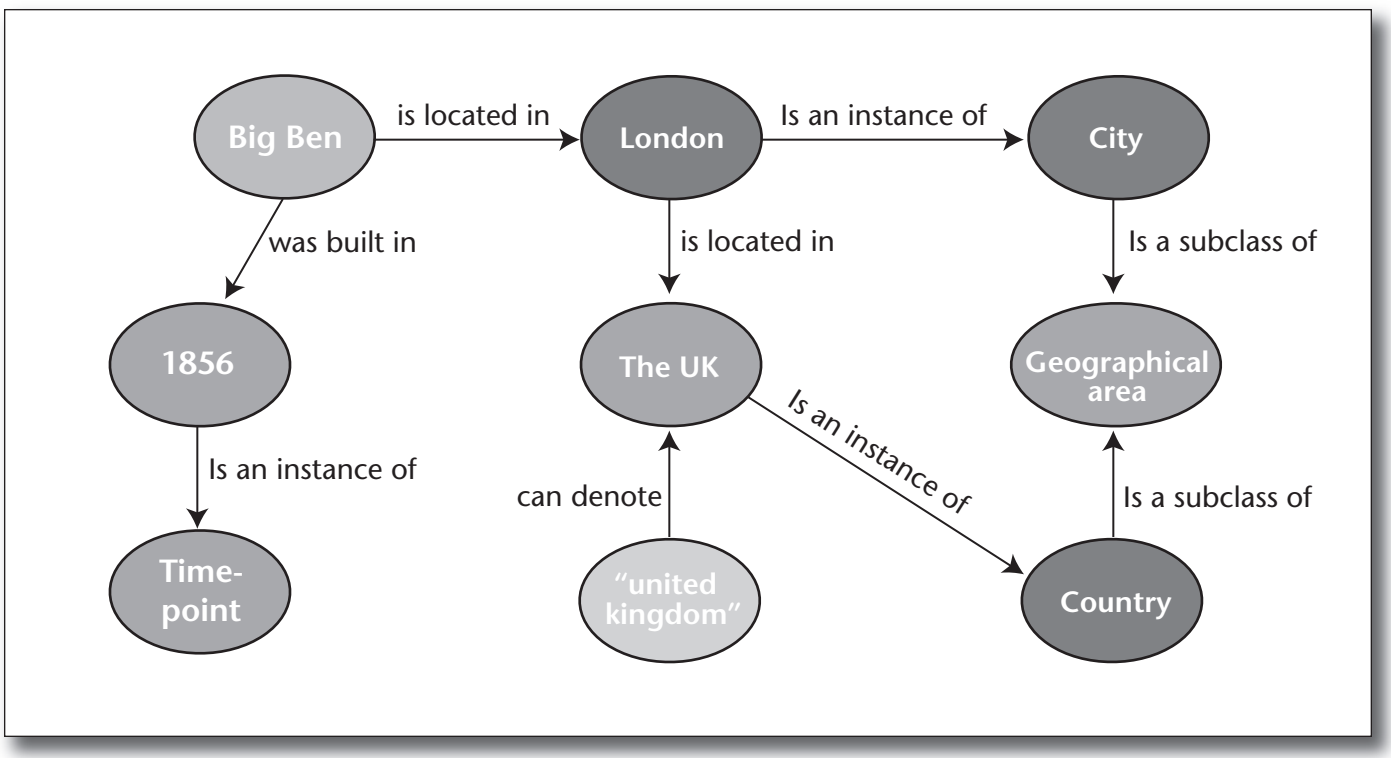

Figure 3. Knowledge Representation.

other facts, saying when they are true, and the temporal properties of various relations and other entities are also stored.

Figure 3 shows a number of facts in graph form. We categorize knowledge into three broad classes. Commonsense knowledge is knowledge that every-

one already knows, such as London being a place, or cities being kinds of place. Although this knowledge doesn't directly answer many useful questions its presence in the platform is vital for making sense of user questions and inferring other useful knowledge. Factual knowledge is knowledge that isn't generally known, such as Big Ben being built in 1856. These facts are frequently used to generate the answers to questions (such as "How old is Big Ben?"). Finally, lexical knowledge is facts about what words and phrases are used to communicate various entities in the knowledge base in natural language. The fact that the string "the uk" can denote the United Kingdom is an example. Other types of lexical knowledge supporting various grammatical constructions in English are also used. Lexical knowledge is vital for question answering and to support the understanding of the millions of ways that questions can be phrased. Approximately a third of the knowledge in the True Knowledge knowledge base is lexical, another third common sense, and the remaining third factual.

All True Knowledge entities have an ID that is typically written in square brackets such as [barack obama], [integer: ["26"]], [is married to]. Note that although, for convenience, English words and phrases are used for the IDs they are actually language-independent identifiers. An alternative implementation could just as easily use numerical IDs.

For an entity to be fully supported in True Knowledge it needs a number of things to be associated with the ID. The first is what we call a *unique recognition string* (a URS). This is a noun phrase that unambiguously describes the entity and that would enable anyone familiar with the entity to recognize it. Examples include "Cambridge, the city in England" and "Barack Obama, the 44th president of the United States." In addition, each entity has a *common translation* that is a shorter preferred noun or noun phrase for the entity (for example, "Cambridge, England," "Barack Obama"). The system also requires knowledge of which classes the entity belongs to and a minimum amount of lexical knowledge for the entity so that it can be asked about in questions. All this knowledge is represented in standard format as facts.

Ontology

One common example of commonsense knowledge that is used heavily is facts that support the classification of entities and the relationships between classes. The main part of the ontology (excluding animal and plant species and products) has more than 20,000 classes.

The requirements for our ontology were that it needed to cover and describe all entities that could be talked about and be strongly semantic with each class well defined and with the facts about each class being robust. There was no external ontology available that had these requirements so we built our own by hand. Although hand creation of knowledge potentially runs counter to the design philosophy of the system that it has to scale, it is our belief that the building of such an ontology is a relatively contained problem and the bulk of the work is already behind us. For example, although classes are still being added, the rate of adding new classes has slowed and the vast majority of new entities that get added to the knowledge base already have an appropriate class.

Figure 4 shows a tiny section of the class tree beginning with [track]: long, thin geographical features such as roads, canals, and railway lines. For clarity some classes aren't shown.

This section is a tiny corner of the entire ontology that covers all entities that it is possible to think about or talk about. The entire class tree has as its root the entity [object], which has subclasses including [physical object], [conceptual object], and so on.

As previously mentioned, the ontology is not stored separately in the platform but is represented entirely as facts asserting relationships such as [is a subclass of] between pairs of classes. Because of this unified approach the ontological knowledge is available for question answering in exactly the same way as any other knowledge. An example of this (the response to "is a bat a bird?") is shown in figure 5.

Query Language

True Knowledge queries are language-independent representations of questions. An example query is shown in figure 6.

The query language corresponds closely to the fact representation except that variables can take the place of IDs. There are no primitive values. The header of the query also allows the user to specify what variables are the result of the query. Without any header variables the semantics of the query corresponds to a yes or no question.

In the example query, the query-processing engine has to find a value for the variable *a* that satisfies the other constraints: namely that a fact *f* exists where *a* is the first entity, the relation is [is the president of], and the right entity is the United States. The final constraint is that fact *f* is believed true at the point in time that has the attribute [current time] (represented in the first line by the variable now). Semantically this query is identical to the question "Who is the president of the United States?" Had the query not specified the variable *a* as the result of the query, the corresponding English question would have been "Is there a current president of the United States?"

Inference

Inference is the ability to generate knowledge

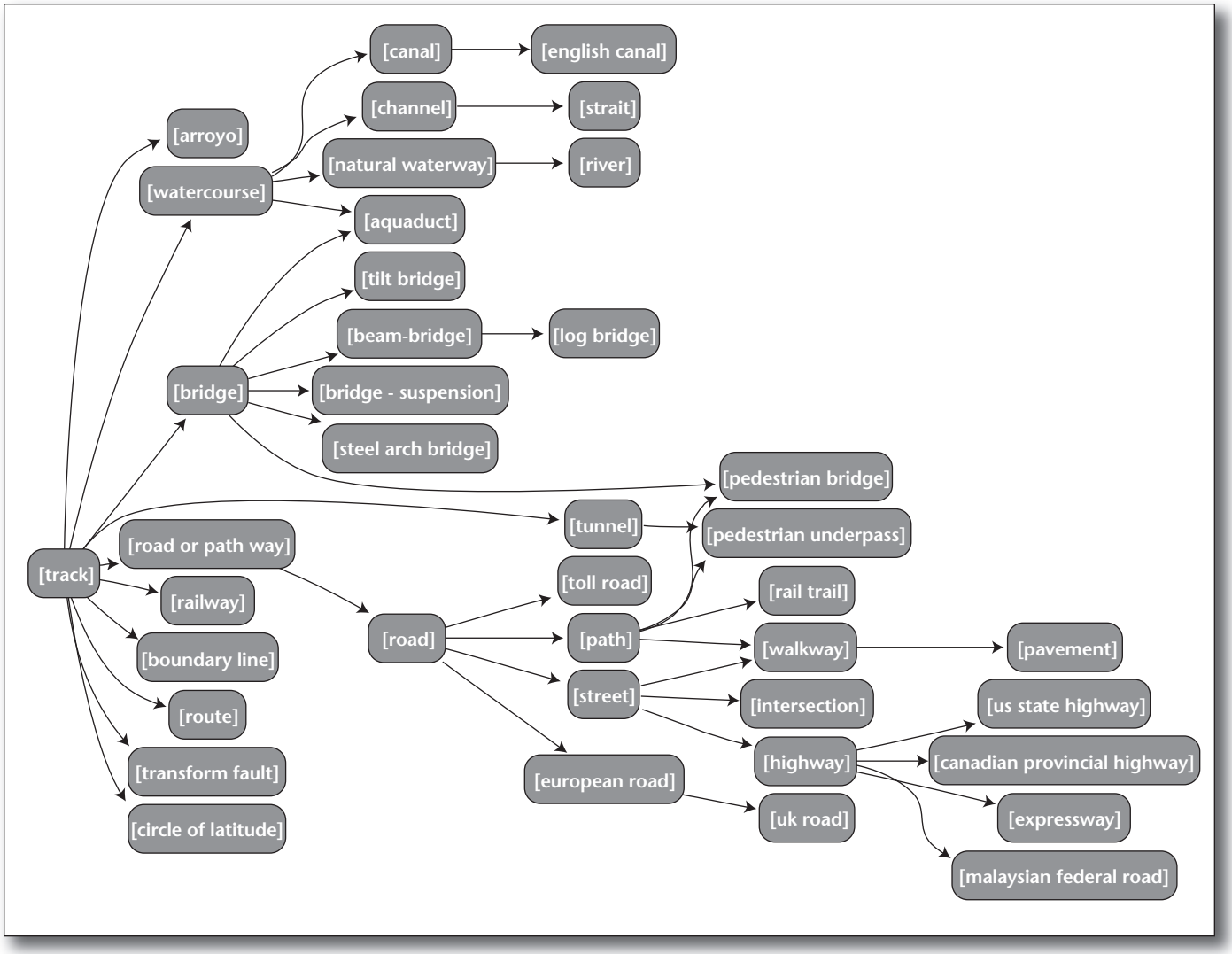

Figure 4. Part of the Ontology.

from other knowledge. We believe that this is an essential capability for any horizontal question-answering system as there is far more knowledge than can reasonably be stored statically. True Knowledge has a general inference system that, while processing a query, generates knowledge as needed from other knowledge (that is, the facts are generated dynamically as needed). A similar system also generates some static knowledge that is available before the query is executed, for performance reasons.

Figure 7 shows a very simple example of inference. From the facts that Big Ben is located in London and that London is located in the United Kingdom it is possible to dynamically generate the fact that Big Ben is in the United Kingdom. No such static fact needs to be stored. (For simplicity the corresponding temporal facts that would also be generated are not shown.)

Inference is implemented by a collection of inference rules, which we call "generators." The current production system contains around 1,500 of them. These are designed to be as general as possible. An example of a generator is shown in figure 8, which implements the concept of a "symmetric" relation. This rule allows the system to infer that Michelle Obama is married to Barack Obama when the only known fact says that Barack Obama is married to Michelle Obama. Most other relations (for example, [is a parent of]) do not have this property. As the True Knowledge system has a temporal model, the rule also asserts that the reverse relation is true for the same periods of time.

Computation is supported too. Some generators have program code attached that can do arbitrary calculations. This program code is independent of the core query-processing engine and could potentially be provided by remote web services. These

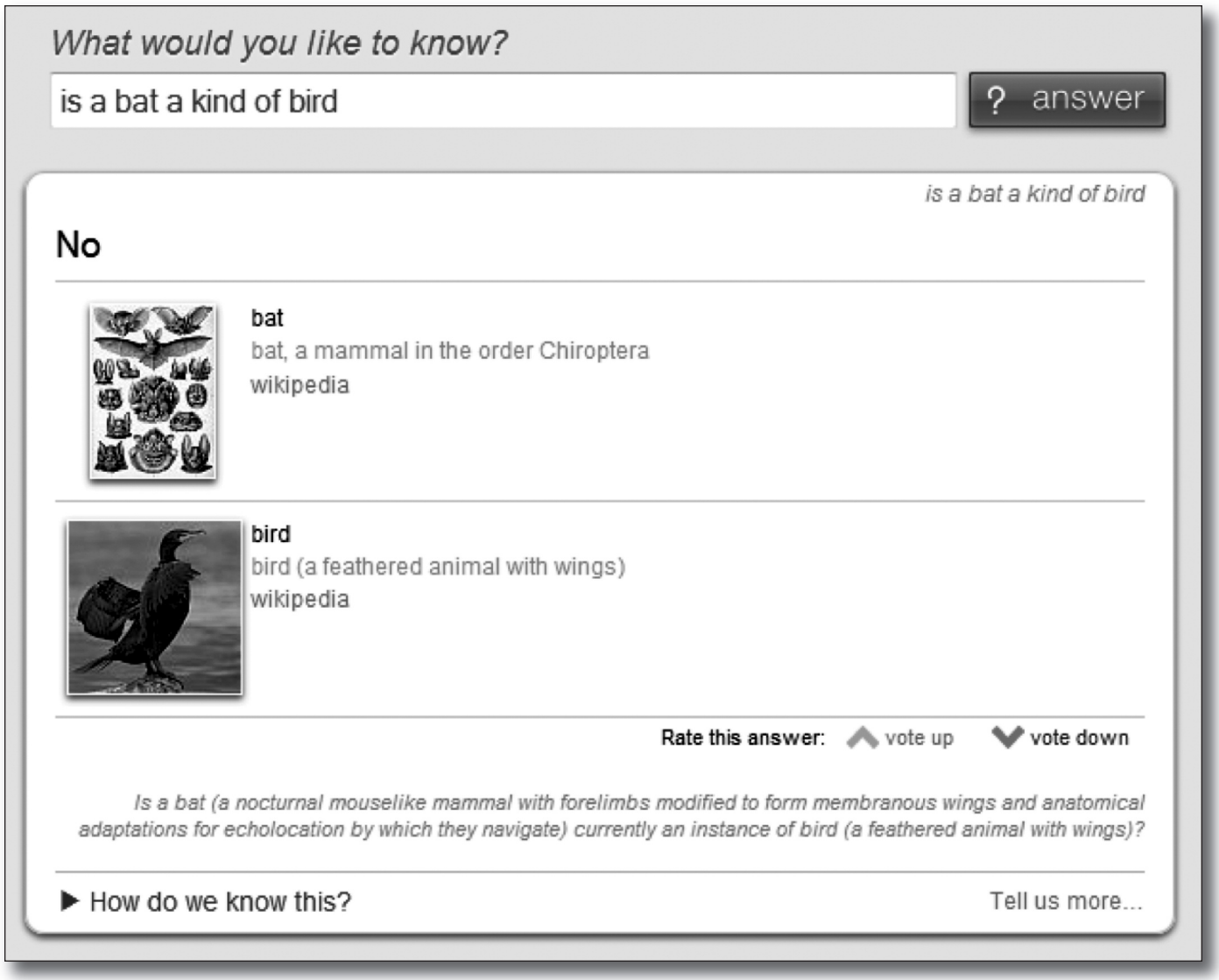

Figure 5. Answering a Question with Ontological Knowledge.

so-called smart generators are reserved for situations where the inference cannot be done by rearranging the results of a query.

An example of a smart generator is given in figure 9. It calculates the day of the week for any given date. This rule enables, for example, the system to answer "Friday" to the question "What day of the week was the 3rd of July 1863?"

Explanation Generation

The query-processing engine is capable of tracing the path it followed to generate answers in order to create a detailed explanation of how those answers were generated. In addition, the static facts used as part of that proof can be extracted and presented to the user as a concise explanation. Our experience is that the concise explanation is usually sufficient, as users find it easy to fill in the inference steps mentally.

```
query a
[current time] [applies to] now
f: a [is the president of] [the united states of america]
f [applies at timepoint] now
```

Figure 6. Example Query.

Translation

Translation is our term for the process of turning natural language questions into the language-independent True Knowledge queries. We implement this with a collection of templates, each of which describes how to turn a class of natural language questions into the correct query. A postprocessing step disambiguates and throws away unlikely interpretations of the question.

A simple example of a translation template is

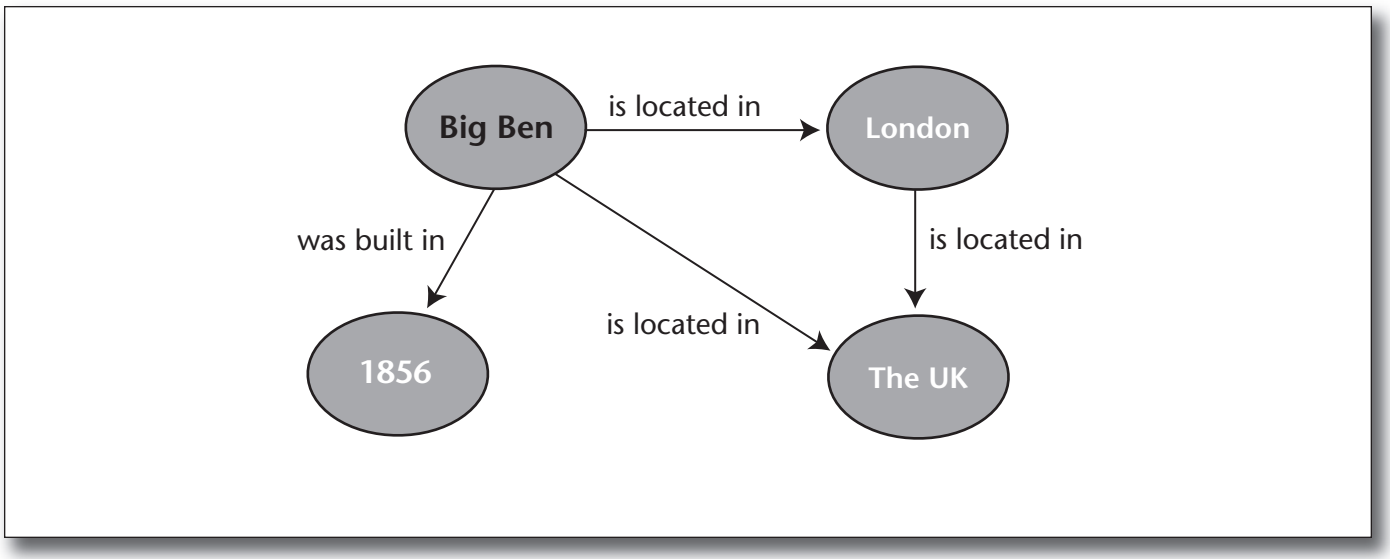

Figure 7. Inference.

```
generator a%,b%,tr
[symmetric] [applies to] r$
f: a% r$ b%
f [applies for timeperiod] tr
=>
g: b% r$ a% *
g [applies for timeperiod] tr
```

Figure 8. A "Dumb" Inference Rule (Generator).

```
generator
day$ [is an instance of] [day of the week]
=>dayofweeksmart@local
[timepoint: string$] [is an instance of] day$ *
```

Figure 9. A "Smart" Generator.

shown in figure 10. The template gives broad instructions as to when and how to translate questions that ask for an unknown object with a named relationship to another object in the present. Examples of questions in this class include "What is the capital of France?" "What is the age of the Golden Gate Bridge?" and "What is the local time in Chicago?"

Translation templates have three main components. The first component is one or more matchlines that are sequences of known and unknown strings. The unknown strings are given a variable name so that they can be referred to later. The current production translation system for True Knowledge has approximately 1200 translation templates and has an efficient mechanism for indexing and rapidly matching a question to all the matchline patterns, and returning the translation templates that match. For example, with the question "What is the capital of France?" the system would match this question to the matchline with the string "is the capital of" being matched to the variable *a* and "France" being matched to the variable *y*.

The next step is to substitute the unknown strings into the header query. This gives the query shown in figure 11, which is then executed and gives the results *a* = [is the capital of] and *d* = [france].

Where words have multiple senses, these queries can produce multiple candidate translations, and in this case a second result is produced where *a* = [is the capital of] and *d* = [france national football team] (which is what "France" would denote in the question "Who is the captain of France?").

These results can then be substituted into the footer query to give two candidate translations of the question, as shown in figure 12.

The system is capable of eliminating additional candidate translations using the commonsense knowledge in the platform.

In the above example the candidate translation asking for the capital of a soccer team is eliminated by knowledge that the "right class" of the relation [is the capital of] is [governed area] (a large category of place) and the system's ability to infer that the French soccer team is not such a place. As a consequence, only one translation survives, and the one remaining query can then be executed to produce the answer.

If more than one candidate translation remains, a number of different regimes are supported. These include "answer combining" — when a unified answer is given if there are only a small number of possible answers. For example, to the question "is Georgia a country" a combined answer could say "if you mean Georgia the U.S. state, the answer is no; if you meant Georgia the country, the answer is yes." Other approaches are to ask users which interpretation they meant, or to pick the most likely interpretation and allow users to select a different one if the selection was incorrect.

System Assessment

One advantage of the inference system is that it can be used to validate knowledge that is coming into the platform. In the event that it is in semantic conflict with other things it knows, the knowledge can be switched off and not used for question answering. We call this *system assessment*. For example, an incoming fact that Barack Obama was born in Chicago can be rejected by using the existing knowledge that he was born in Hawaii, that people only have one place of birth, and that Chicago is geographically distinct from Hawaii (which can be readily determined by inference).

Users are allowed to endorse or contract facts stored in the system, optionally citing additional sources, and this history is stored. If multiple users or sources say that a fact is untrue it can be switched off even if it isn't in conflict with other knowledge. We term the ability for users to endorse or contradict static facts and provide additional sources to back up these assertions as *user assessment*.

When facts are system assessed as false the facts stay in the knowledge base but are labeled as "believed untrue" and not used for answering questions. If knowledge changes and it turns out that it was the original knowledge that was untrue after all (perhaps after user assessment), the switched-off fact will be automatically resuscitated.

Knowledge Acquisition

Knowledge acquisition is one of the key challenges for our approach as questions can only be answered when sufficient knowledge is available to both understand and answer the question.

Knowledge is added from multiple sources and by following multiple strategies. A substantial percentage of the staff inside True Knowledge Ltd are involved in either directly adding knowledge or in developing tools that add knowledge at scale. Each source typically both creates new facts and provides additional sources or endorsements for existing facts. The various sources and strategies include databases, Wikipedia, user-supplied and knowledge extraction using natural language processing.

Databases. Most structured data can be imported into the knowledge base. The process of importing a typical relational database involves mapping the implied relations from the tables in the database to True Knowledge relations, and mapping the contents of the fields to True Knowledge entities to create facts. Our experience is that, once imported, the inference system and commonsense knowledge in the platform make the recently imported knowledge substantially more useful than it was in its original context.

In addition to conventional databases, we use Freebase as a source that provides structured knowledge across many different areas.

Wikipedia. Wikipedia is an extremely successful

Match:
"what"/"which" a y

Header:
query r,d
a [is a present central form of] r
y [can denote] d

Translation:
query b
[current time] [applies to] now
f: b r d
f [applies at timepoint] now

Figure 10. An Example Translation Template.

query r,d
["is the capital of"] [is a present central form of] r
["france"] [can denote] d

Figure 11. Translation Step One.

query b
[current time] [applies to] now
f: b [is the capital of] [france]
f [applies at timepoint] now

query b
[current time] [applies to] now
f: b [is the capital of] [france national football team]
f [applies at timepoint] now

Figure 12. Candidate Translations.

online encyclopedia that contains significant knowledge on most notable subjects. Although mostly unstructured, the English version of the website contains many semistructured elements such as summary tables ("infoboxes") and category information. Within True Knowledge we have a software system that mines knowledge from this source and keeps this knowledge up to date as Wikipedia pages are added and changed. We are also beginning to extract significant knowledge from the unstructured parts of the pages using natural language processing, as will be described.

User Added. One other source of knowledge is user-supplied knowledge: facts added by people interacting with the trueknowledge.com website.

In many cases when a question is understood but the answer is not known, the user can be prompted for the missing factual knowledge, which is then available to answer the original and other questions.

Additionally, the translation system described above can translate user input to fact assertions. Users typing in natural language assertions such as "Barack Obama was born on the 4th of August, 1961" can thereby express the fact they wish to add directly.

An example of a user adding knowledge is shown in figure 13. The user has asked the question "How tall is Adrianne Palicki?" The question was understood by the system but the answer was not known. With the question translated correctly, the system was able to identify the missing knowledge and provide the user with a link to add it. The user then clicked the link to add the missing knowledge and was guided through the process of adding the missing fact by the system.

Less than one fact in a thousand in the knowledge base has been supplied by users outside the business. However our experiments have shown that this source is disproportionately important with more than 10 percent of the questions we answer using at least one user-added fact to generate the response.

Natural Language Processing. A recent direction for the company is extracting facts from unstructured web pages using natural language processing (NLP). The basic process is simple and builds heavily on our current technology. Our four-stage process of sentence extraction, simplification, translation, and bootstrapping allows us to extract high-quality facts that don't degrade the overall quality of our knowledge base. We begin by crawling the web for sentences stating assertions, and then simplify these assertions into the format "subject-noun-phrase verb-phrase object-noun-phrase."

These simple sentences are then translated by the True Knowledge translation system into facts. The approach is identical to the method used to translate questions except that the output is one or more facts instead of queries. Finally, facts extracted by our web crawler are boot-strapped into our knowledge base. This boot-strapping process evaluates the likelihood of a fact being true based on facts we already know, the track record of facts extracted by the web crawler about the same objects and relations, and the track record of facts extracted from the same site.

True Knowledge's technology is leveraged twice in this process. Initially we use our translation technology to translate and disambiguate the simplified sentence into a fact. System assessment is then used during the boot-strapping phase where we check that the fact makes sense and is consistent with our world view. For example, given the sentence "David Letterman is also a television and film producer," we will simplify it into "David Letterman is a television" and "David Letterman is a film producer." These are translated into two facts: [david letterman] [is an instance of] [television] and [david letterman] [is an instance of] [film producer]. During boot-strapping the first fact will be discarded as the system can infer that no people are televisions. In contrast, the second fact is consistent with our world view and is kept.

Accuracy is the priority in our NLP extraction. Due to our stringent filtering only small percentage points of simplified sentences are translated into unambiguous facts, and only a small minority of those are boot-strapped onto the knowledge base. The advantage of this cautious approach is that we are seeing accuracy rates of 98 percent for facts extracted from the web using NLP with no curation.

Vertical Areas

The True Knowledge technology is fundamentally horizontal: it was designed to support knowledge across all knowledge areas simultaneously and to understand and answer questions on all subjects.

However, this doesn't mean that the technology cannot be used to support vertical applications. It can — simply by comprehensively fleshing out a corner of the knowledge base.

One vertical area that has been worked on is product/local search, answering questions in the class "where can I buy a <named product> in/near <named place>?" Multiple inference paths are supported, linking types of products, relationships between products, relationships between products and types of retailer, and retailer data (facts about retailers and where they are). At the time of writing, True Knowledge has added details to its platform of some 325,000 retail businesses in the UK. These answer natural language questions in this area, but can also be used (through an API) to power applications that are *only* interested in this knowledge area.

Figure 14 shows a web application[1] that does queries of these types. The application is powered

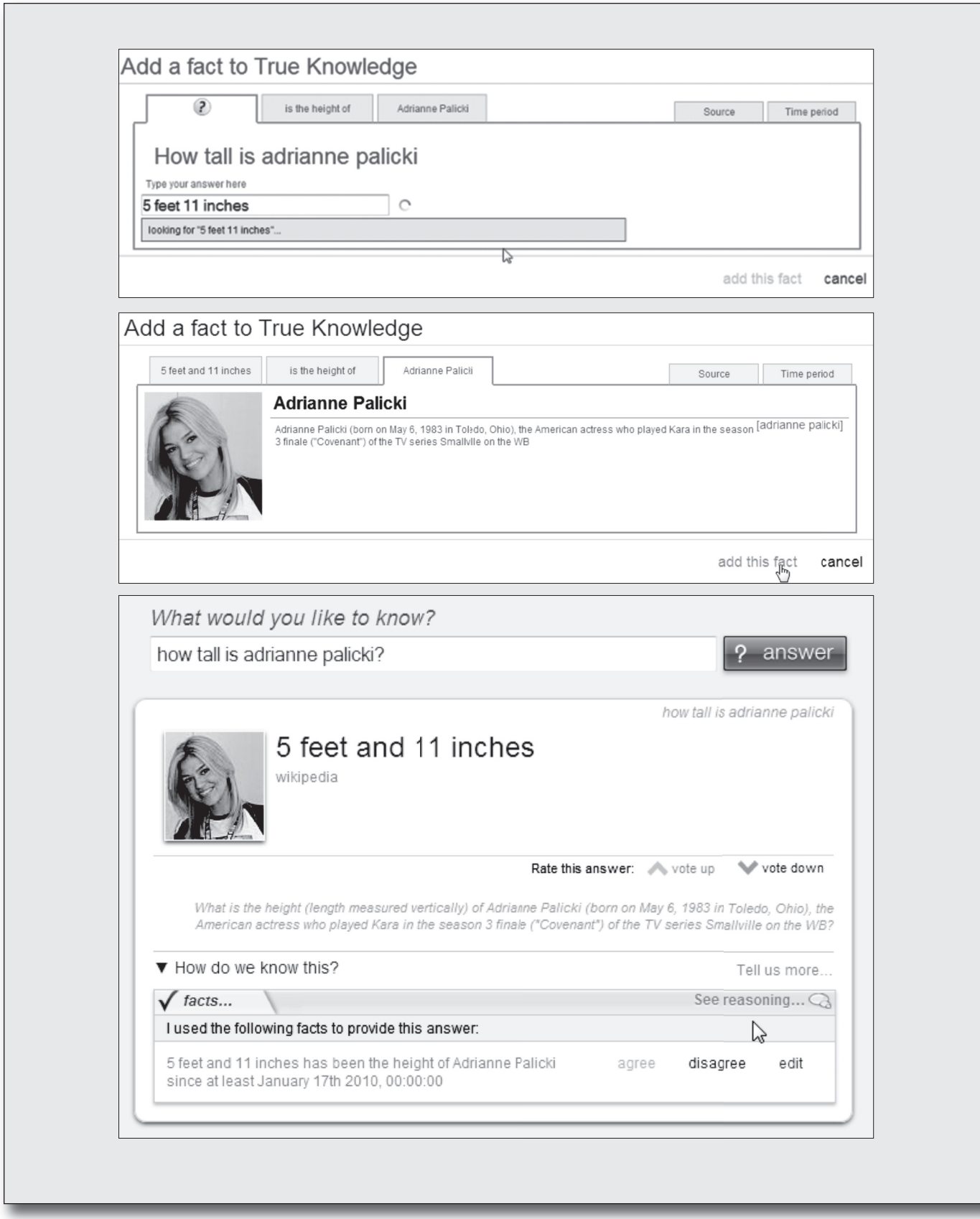

Figure 13. User-Added Knowledge.

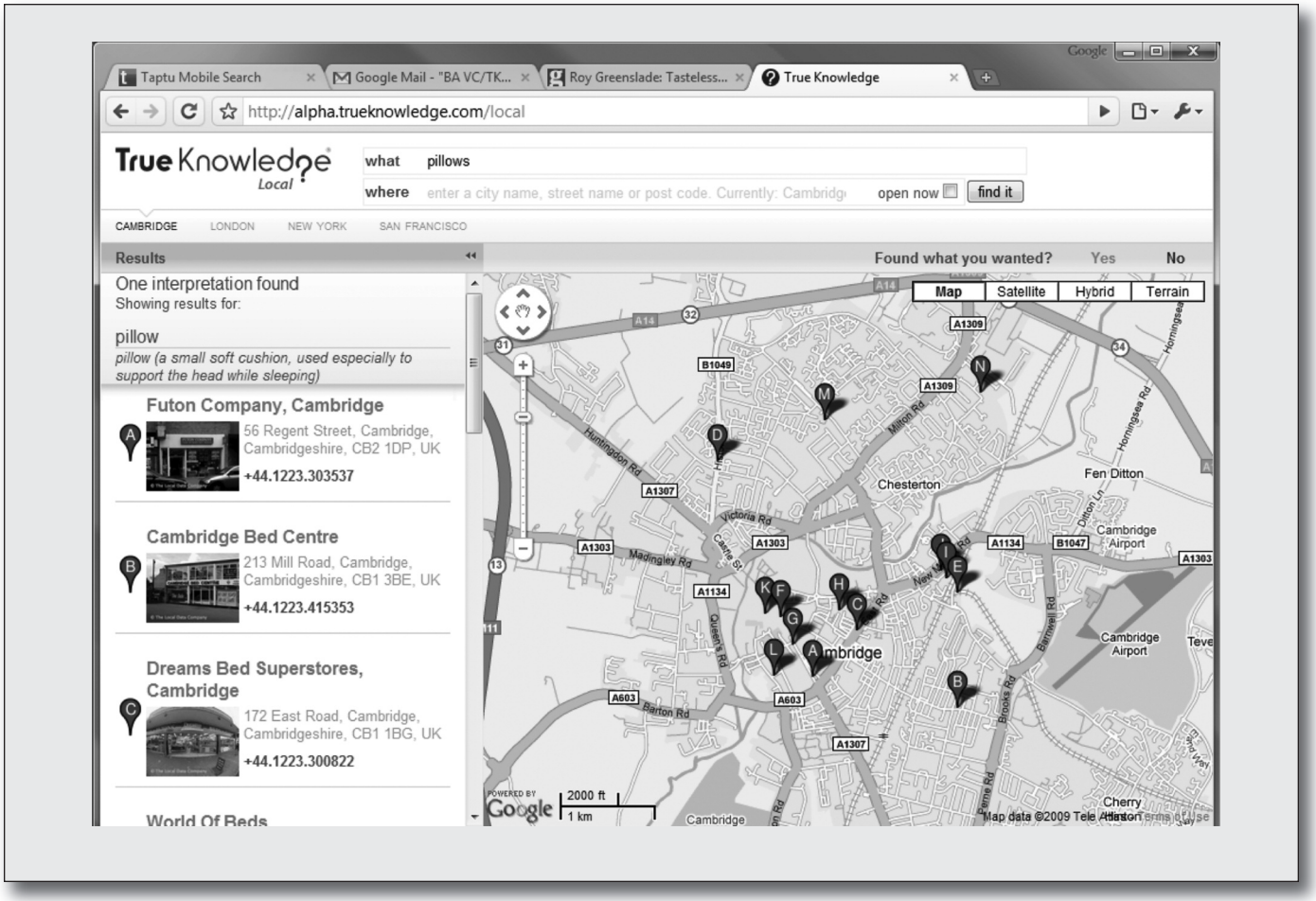

Figure 14. Product Search Application Built on Platform.

by the main True Knowledge platform but bypasses the translation layer to query the knowledge base directly for retailers in a named geographical location. The platform is also queried for each retailer's contact details and location. The results are then shown graphically on a map.

Image Selection

A valued part of the user experience when interacting with the True Knowledge website is the automatic display of images appropriate to the results.

Images are entities, just like any other entity in True Knowledge, and facts exist that say what the images depict. The retrieval of the actual image is done by a relation that links the image entity to a URL where the image is stored. In February 2010 we are approaching a million high-quality images associated semantically with entities.

If an image exists for the answer or answers to a question this is the ideal. However, as the question is semantically understood the system also knows the entities involved in the question. In the absence of an image of the answer, the response can be illustrated by an appropriate image relating to an entity in the question.

Comparison with Other Systems

Although True Knowledge's origins are separate from other projects it has similarities with other systems.

Wolfram Alpha is possibly the closest analogous system to True Knowledge, combining vast amounts of structured data and computation to respond directly to user queries. However, there are some significant differences in approach. One of the main ones is Wolfram Alpha's use of program code to bring new knowledge areas into the system. Wolfram Alpha's knowledge is stored in a large collection of separate database tables, and its functionality comes from some 6 million lines of Mathematica code.[2] In contrast, True Knowledge has a single knowledge base representing all the knowledge it knows in a unified format. Queries are solved in a knowledge neutral manner as the query processing system contains no code relating to any specific knowledge vertical.

Another difference is the emphasis on semantics.

All True Knowledge entities fit into a unified ontology while Wolfram Alpha has no ontology.[3] True Knowledge also attempts to create a full semantic map between all the words in the user's question and a corresponding query, while Wolfram Alpha's main mechanism for processing a question involves extracting and matching the main words in the user input without any complex parsing.[3]

The role of curation in knowledge acquisition is another difference in approach. Wolfram Alpha has a large team of people involved in knowledge acquisition and curates all data before putting it into the system, even hiring domain experts prior to starting work on a new vertical area. In contrast, much of the knowledge that True Knowledge acquires comes from automatically mined sources and users, without any curation, and the system attempts to maintain the quality of this knowledge using automatic methods such as the system assessment and user assessment systems described previously.

Freebase[4] is a project to compile a large structured knowledge base of the world's knowledge, constrained to knowledge that can be made available under a free Creative Commons license. Like True Knowledge, the platform has an API that automated systems can query. However, there is no natural language question-answering ability (though Powerset has applied its natural language capabilities to Freebase data). Another difference is the lack of an inference system in Freebase. Topics are also grouped into broad top-level categories rather than into a full ontology.

CYC[5] is an AI project that was started in 1984 and, like True Knowledge, combines an ontology, structured representation of commonsense knowledge, and an inference system. One of the main differences in core technology between CYC and True Knowledge is the complexity of the underlying knowledge representation system: CYC's knowledge representation language CycL combines first-order logic with modal operators and higher-order quantification.[6] Another difference is that CYC groups all its knowledge into "Microtheories" that correspond to a realm of knowledge. Consistency of knowledge is only required within a Microtheory.[7] In contrast True Knowledge attempts to maintain consistency across all its knowledge.

Progress and Results

Figure 15 shows the growth in the knowledge base. In February 2010 the system has 240 million facts about 8 million entities. The original prototype system was developed with a few hundred facts so we have already scaled the system through six orders of magnitude. If we succeed in our ambitions, we hope to grow what the system knows by several more orders of magnitude.

Question-Answering Capabilities

Evaluating the question-answering capabilities of a system like True Knowledge is complicated by the fact that question streams from different sources have very different characteristics. At one extreme we have freely typed questions asked directly to the True Knowledge platform by users, which include people who have some familiarity with the system and its capabilities. We are automatically answering much more than half of these questions. At the other extreme are questions sourced from discussion forums or other sources where the person asking the question was expecting a person to answer, and consequently there is no incentive to phrase the question well. Such sources frequently have poor spelling and grammar and often lack the necessary context to understand the user intent. These can be further complicated when the set of questions from the source is biased towards more difficult questions due to difficulty finding the answer from sources such as web search. For the worst of these sources True Knowledge answers around 2 percent though we have seen human-to-human sources where the system answers close to 10 percent.

A concept that the author finds useful is a hypothetical benchmark that we call "answerable life questions" (ALQs). This benchmark is defined by the day-to-day information needs of a population of users (the English-speaking world, say). The hypothetical situation is that every time someone in that population needs some information that they don't already know, they ask for that information with a naturally phrased, but well-constructed question (that is, not necessarily grammatical or correctly spelled but one understandable by a native speaker, and having enough information in the question to reliably infer the user intent). The benchmark is the collection of all these questions, further limited to those that are reasonably answerable (that is, a well-resourced person could find the answer from an online source without extraordinary effort). Although no such benchmark exists, we have made efforts to construct proxies for it using a mixture of sources and some guesswork, and testing shows that we are currently answering around 17 percent of the set. Sampling of the unanswered questions in the benchmark shows that about a further 36 percent could be answered simply by the addition of new facts to the knowledge base, and roughly 20 percent more could be answered by adding appropriate translation templates or generators. The remaining questions require further extensions to the technology or need to be answered with a prewritten editorial response (for example, "How do I remove the battery from my iPhone?"). Our technology can be used to match such editorial responses to questions and their

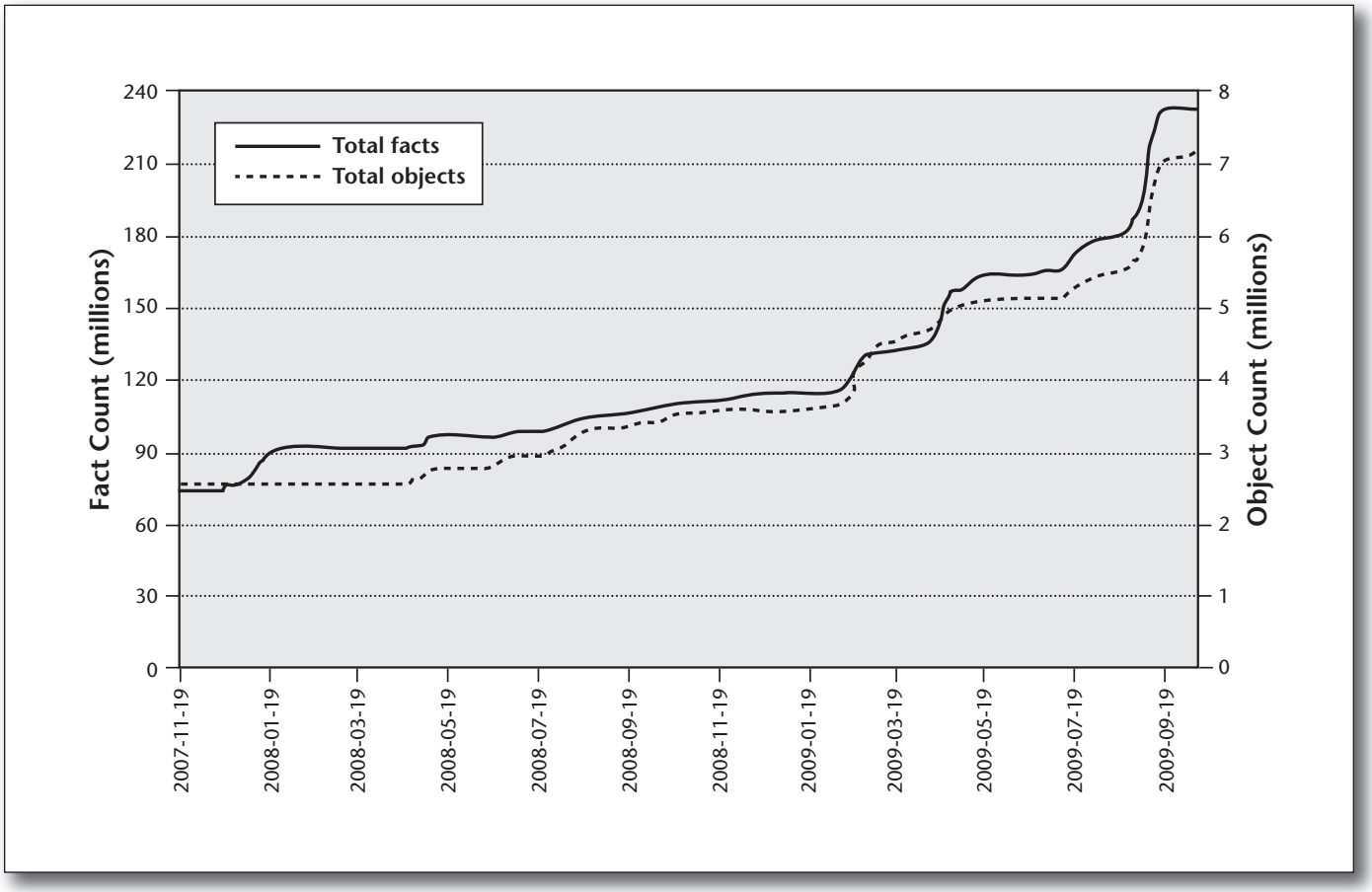

Figure 15. Growth in Facts and Objects.

variants, but scaling such responses presents some difficulties.

The Future

One advantage of the True Knowledge approach is the language independence of the technology.

Although the current implementation supports only English, all the natural language capabilities are limited to the translation system, and, with the exception of some smart generators that support English-language lexical relations (how to pluralize a noun for example), there is no English-specific program code in the platform. All the English-specific components are limited to the collection of translation templates and the lexical knowledge. The commonsense and factual knowledge is stored in language-independent form, and the query-processing and inference system is also language independent.

Supporting another language is therefore a matter of creating a new set of translation templates for the target language and adding appropriate lexical information for the concepts where the knowledge is different (most of the current lexical knowledge relates to proper nouns that are often the same across languages). With such a system implemented, users would be asking questions to the platform in multiple languages but having the answers generated with shared factual knowledge.

Notes

1. See local.trueknowledge.com.
2. See Wolfram Alpha official website: www.wolfram.com/news/wolframalpha.html.
3. Stephen Wolfram, personal communication (2009).
4. See www.freebase.com.
5. See www.cyc.com.
6. See www.cyc.com/cycdoc/ref/cycl-syntax.html.
7. See www.cyc.com/cycdoc/course/what-is-a-context.html.

William Tunstall-Pedoe is the founder of True Knowledge and the inventor of the technology. Tunstall-Pedoe is a Cambridge University computer science graduate. His career has been spent developing AI applications and marketing them through businesses he has founded. Previous products include a commercial chess-playing program, the first and only program that can solve and explain cryptic crossword clues, and the AI anagram-generating software used by Dan Brown to create the anagrams that appeared in the *Da Vinci Code* book and movie.