Combining Information Seeking Services into a Meta Supply Chain of Facts

Dmitri Roussinov
Department of Information Systems, Arizona State University
Email: dmitri.roussinov@asu.edu

Michael Chau
School of Business, The University of Hong Kong
Email: mchau@business.hku.hk

Abstract

The World Wide Web has become a vital supplier of information for organizations in order to carry on such tasks as business intelligence, security monitoring and risk assessments. Having a quick reliable supply of correct facts from the outside environment is often mission critical. By following the design science guidelines we have explored ways to recombine facts from multiple sources, each with possibly different responsiveness and accuracies into one robust supply chain. Inspired by prior research on keyword based meta search engines (e.g. metacrawler.com) we have adapted the existing question answering algorithms to the task of analysis and triangulation of facts. We present a first prototype for a meta approach to fact seeking. Our meta engine sends a user's question to several fact seeking services that are publicly available on the Web (e.g. ask.com, brainboost.com, answerbus.com, NSIR etc.) and analyzes the returned results jointly to identify and present to the user those that are most likely to be factually correct. The results of our evaluation on the standard test sets widely used in prior research support the evidence for the following: 1) the value-added of the meta approach: its performance surpasses the performance of each supplier, 2) the importance of using fact seeking services as suppliers to the meta engine rather than keyword driven search portals, and 3) the resilience of the meta approach: eliminating a

single service does not noticeably impact the overall performance. We show that those properties make the meta-approach a more reliable supplier of facts than any of the currently available stand-alone services.

Keywords: question answering, fact seeking, meta search, business intelligence

Introduction

Modern organizations have to stay aware of the increasingly more dynamic environments in which they operate. The World Wide Web has become an important supplier of information which it can provide in a number of ways. For example, technical personnel regularly search for solutions to common problems. The Web supplies facts about competitors and partners, news articles, stock trends, customer perceptions, company backgrounds, prices of services and their availability. In becoming flatter and more global operational landscapes, the information captured in Web pages allows organizations to cross the borders virtually into other countries and cultures, thus opening new markets and exploring new opportunities. Web search engines are commonly used to locate information by business analysts (Chen et al., 2002; Chung et al., 2005; McGonagle and Vella, 1999). That is why it is not surprising that the Web portals Google and Yahoo together rivaled the prime time advertising revenues of America's three big television networks, ABC, CBS, and NBC (The Economist, 2005).

In this work, we start from considering the Web as a giant information supply chain (or a network in a more general case). A conceptual diagram is depicted in Figure 1. Millions of facts are posted online daily embedded in Web pages created by individuals and organizations. Those pages are crawled by search portals to create gigantic databases

(indexes) of all the publicly accessible pages to allow fast retrieval of those that match user-supplied keywords. When receiving a keyword query, Web search engine portals like Google or Yahoo typically retrieve a large number of pages and overload business analysts with irrelevant information (Chen et al., 2002). While making large advances in the ability to find the most popular Web pages containing user's keywords, Web search portals are still not designed to deal with fact seeking tasks. Instead, they treat the tasks (questions) as simple keyword queries ("bags" or "sequences" of words). For example, when a user types a question "Who is the largest producer of software?" into MSN search engine, it is treated in almost the same way as if the user typed "software producer largest" resulting in large[1] overlap between the top 10 pages returned as a response. To disorient the user even further, the returned pages mention "largest producer of insulators," "spam producers," and "custom calibration software," but not the answer that the user would be expecting (for example, "Microsoft" at the time of writing this paper). At the same time, previous research has noted that a significant proportion of queries on search portals have a specific question in mind even if the query was not entered as a question (Radev et al., 2001; Radev et al., 2005).

[1] 50% at the time of the study

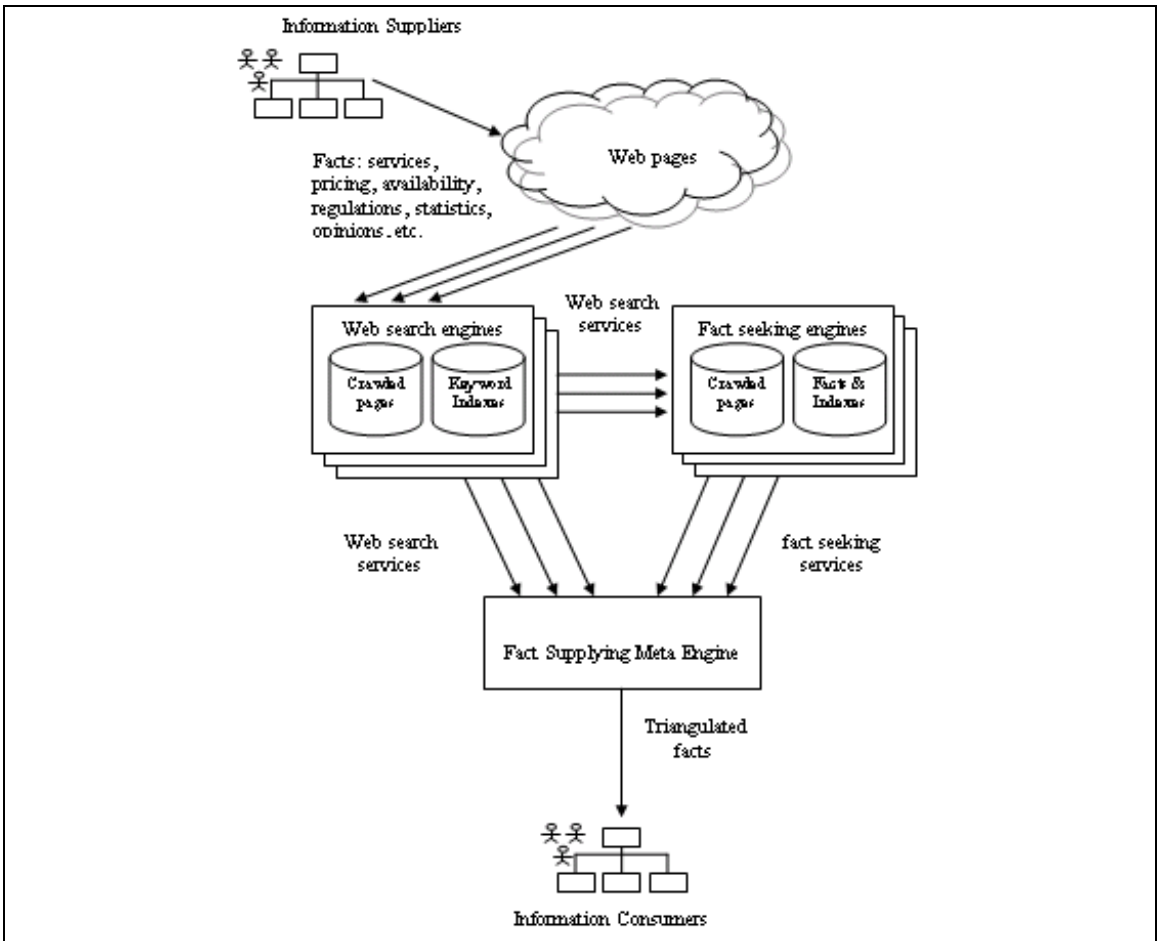

Figure 1. The Web, Search Portals and Meta Fact Seeking Engine as an Information Supply Chain

In this work, we consider specifically a supply chain of facts that can be requested by stating a question in natural language. Question Answering (QA) technology stands behind the automated fact seeking process (Voorhees, 2003). QA serves to locate, extract, and represent a specific answer to a user question expressed in natural language. For example, a QA system takes an input such as “How many cars are sold in Turkey?” and provides an output such as “In Turkey 2,000 to 3,000 vehicles were sold yearly”, or simply “2500”.

Although the correct answers can be frequently “eye-balled” within the snippets provided by keyword driven search portals with response to a carefully crafted query or even the entire question entered verbatim, in some very important situations the eye-balling approach is not adequate due to time crunch or communication bandwidth limitations. For example, a growing number of mobile device users do not have the luxury of a large screen space to make eye-balling quick. Military or first-responder systems require accurate answers within seconds in order to minimize risk to human lives. Visually impaired computer users cannot glance through pages of snippets and would certainly benefit from having a compact answer which the current speech-to-voice technology can convert into audio output.

Since the support for the types of questions going beyond simple fact seeking (e.g. for questions expecting common sense reasoning) by the online services is still very limited, we focused our current study on factoid (fact seeking) questions. In the study reported here, *we have explored a specific approach to creating a reliable and fault-tolerant supply chain capable of delivering facts stated anywhere in the entire Web, doing this automatically and on demand.* The facts are gathered in real-time from various services on the Web capable of responding to questions expressed in human language, analyzed together and presented to the information consumers located higher within that information supply chain.

While following the guidelines of Design Science Research (Hevner et al., 2004) we have accomplished the following: 1) We have critically analyzed the existing technological solutions behind online fact delivery. 2) By following the example of meta search engines on the Web (e.g., the Metacrawler (Selberg and Etzioni, 1995)), we have

suggested an innovative approach to combining several *fact seeking services* (formally defined below in the "Defining the Meta Approach" section) into a single *meta supply chain of facts* (also called "meta-engine" throughout our paper), which sends a user's question to several fact seeking services that are publicly available on the Web (For example, ask.com, brainboost.com, etc.) and analyzes the returned results jointly to identify those that are most likely to be factually correct.

We present a first (up to our knowledge) prototype that exemplifies our proposed meta approach to fact seeking. We demonstrate its added value through batch-mode simulation while testing on a set of questions widely used in prior research. Specifically, we demonstrate 1) *value-added of the meta approach*: its performance surpasses the performance of each contributing service, 2) *the importance of using fact seeking services* for the task discussed here rather than (or in addition to) keyword-driven search portals, and 3) *resilience*: eliminating a single service does not impact the overall performance.

A meta supply chain of information (facts) considered here can be used by organizations in a number of ways, for example to determine what services the competitors provide and at what prices. While shipping to business partners, companies can use it for address verification, and finding about shipping rates or pick-up services. Even simple, common sense facts, can be used to automatically federate heterogeneous databases (For example, if the chain reports that *red* is a *color*, then the one database column containing attributes such as *red, green* and *blue* can be automatically matched to a column called *item color* in another database). Fact supply chain could also be used to find a particular vendor of raw materials and additional information about the vendor,

such as if it is involved in any litigations, government scrutiny or has been subjected to consumer advocates' warnings.

The next section overviews the prior work in the domain. It is followed by the section introducing the prototype and the section on its evaluation. Finally, we conclude our paper with the summary of our findings and our discussion of the limitations and possible directions for future work[2].

Prior Work

Recent Trends in Automated Fact Seeking Technology

The National Institute of Standards and Technology (NIST, USA) has been organizing the annual Text Retrieval Conference (TREC) since 1992, in which researchers and commercial companies compete in such tasks as document retrieval and filtering (Voorhees and Buckland, 2006). The performance of each research team at the competition has significant impact on the government funding of their research efforts. For the last few years, the conference and the funding agencies' priorities have shifted to novel applications, such as question answering, novelty and topic detection, summarization, and interactive Web searching. The participating systems are expected to find exact answers to the so called "factual" questions (or "factoids," such as *who, when, where, what,* etc.), list questions (e.g., *What companies manufacture rod hockey games?*) and definitions (e.g. *What an audit is?*).

---

2 Our results presented here expand and build on our preliminary and less detailed results that appeared earlier in conference proceedings.

In order to answer such questions, a typical system would: (a) transform the user query into a form it can use to search for relevant documents, (b) identify the relevant passages within the retrieved documents that may provide the answer to the question, and (c) identify the most promising candidate answers from the relevant passages. Most TREC QA systems are designed based on techniques from natural language processing (NLP), information retrieval (IR) and computational linguistics (CL). For example, Falcon (Harabagiu et al., 2000), one of the most successful systems, is based on a pre-built hierarchy of dozens of semantic types of expected answers (*location, city, street, profession, person, celebrity, musician, violinist, etc.*), complete syntactic parsing of all potential answer sentences, and automated theorem proving to validate the answers.

In contrast to the NLP-based approaches that rely on laboriously created linguistic templates, "shallow" approaches that use only simple pattern matching have been successfully tried, e.g. the system from InsightSoft (Soubbotin and Soubbotin, 2003) won the 1st place in the TREC competition 2002 and the 2nd place in 2001 TREC. However, none of the best performing systems, including "knowledge heavy" ones such as Falcon and "pattern based" ones such as the one from InsightSoft, is publicly available for independent evaluation or for inclusion in a research prototype. On the other hand, the algorithms behind many of the non-linguistic ("knowledge light") systems have been disclosed (e.g. Voorhees and Buckland, 2006) and are possible to replicate. This may explain why the proportion of participating teams relying on non-linguistic approaches has grown from 12% in 1999 to 76% in 2006 (Voorhees and Buckland, 2006).

World Wide Web as a Source of Answers for Fact Seeking Tasks

There are several important distinctions between factoid question answering from a closed corpus (such as corporate repositories or those used in a TREC competition) and fact seeking from the entire Web studied here, which is typically referred to as *open corpus*:

1) Since the Web has a much larger number of documents (several billion) than a closed corpus (a million or less) has, the former has a much larger variation in the ways in which the answers can be stated, including complex ways (e.g. *On New Year's Eve of 2000, the Eiffel Tower played host to Paris' Millennium* for the question *Where is the Eiffel Tower located?*) or simple ways (e.g. *The Eiffel Tower is located in Paris.*). The presence of answers stated in less complex ways allows the open corpus fact seeking systems to go for "the most low hanging fruit": look for the most easily identifiable answers, making the task very often much easier than a search in a closed corpus (e.g. company repository), and thus not requiring deep NLP processing. This makes open corpus (Web) fact seeking an attractive target for non-linguistic approaches.

2) The users of the Web fact seeking engines do not necessarily need the answers presented stand-alone. In fact, before this study, we had found from interviewing business analysts (recruited among our MBA students) that they prefer to read the answers with the surrounding sentences in order to be more certain in the correctness of the answer. Thus, it is more important for an open corpus (Web) fact seeking engine to recognize the sentences containing correct answers and present them to the user, rather than the verbatim answers, which may be required for applications not involving human users (e.g. automated reasoning).

3) Web fact seeking engines need to be quick to support interactivity, while TREC competition does not impose any real time constraints. This makes simple non-linguistic approaches not only applicable but also the preferred choice over "deep" linguistic analysis.

Those differences shape the design decisions while porting and adapting existing fact seeking techniques to the much larger context of the World Wide Web. AskJeeves (www.ask.com), a public company, positions itself as the pioneer of Web fact seeking. However, their knowledge sources are limited to a small set of specially crafted databases (e.g. geographical locations). When answers are not found there, AskJeeves reroutes the question as a simple keyword query to a general purpose search engine (Teoma, www.teoma.com/). Although AskJeeves recently introduced the "Web answer" automated question answering functionality it still affects only a relatively small proportion of questions (5% in our tests described below).

The Natural Language Processing (NLP) task, which is behind fact seeking technology, is known to be Artificial Intelligence (AI) -complete (Marcus, 1995): it requires computers to be as intelligent as people, to understand the deep semantics of human communication, and to be capable of common sense reasoning. Regarding this, current systems have different capabilities. They vary in the range of tasks that they support, the types of questions they can handle, and the ways in which they present the answers. While looking for answers, users have to switch between several systems, and start their search all over again each time. Beginners can easily get disoriented. They do not have adequate knowledge to decide what system to try first and where to go if that system fails.

START (Katz, 1997; Katz et al., 2004) was one of the first QA systems available online since 1993. It was primarily focused on encyclopedic questions (For example, about geographical locations) and used a precompiled knowledge base. Our experience with the system indicates that its knowledge is rather limited, e.g., it fails on many questions from the standard test sets (detailed below). Mulder (Kwok et al., 2001) was the first general purpose, fully automated fact seeking system available on the Web. It worked by sending user questions to a general purpose search portal (Google), then retrieving and analyzing the returned Web pages to select answers. When evaluated by its creators using TREC questions, Mulder outperformed AskJeeves by a large margin. Unfortunately, Mulder is no longer available on the Web for a comparison.

Radev et al. (2001) presented a relatively complete, general purpose, Web based fact seeking system called NSIR. Dumais et al. (2002) presented another open domain Web fact seeking system (AskMSR) that applies simple combinatorial permutations of words (so called “re-writes”) to the snippets returned by Google and a set of 15 handcrafted semantic filters to verify seven possible categories to achieve striking accuracy. Their work followed the work by other researchers on using the inherent redundancy (repeating answers) on the Web (e.g. Clarke et al., 2001).

The prototypes based on Web fact seeking technologies have been demonstrated to surpass human performance in answering trivia questions (e.g. from “Who Wants to be a Millionaire”) (Lam et al., 2003) and solving crossword puzzles (Castellani, 2004). Roussinov and Robles (2005) studied how automated open domain (Web) question

answering can facilitate business intelligence tasks and the task of locating malevolent online content within cyber security applications (Roussinov and Robles, 2007).

The Approach Studied: Meta Supply Chain of Facts

Defining the Meta Approach

A single portal can play a role of a meta engine: it can send a user's question to several publicly available fact seeking services (e.g. AskJeeves, START, NSIR, etc.), then analyze and combine the results. We define a *fact seeking service* to be supplier of *candidate answers* to at least *some types of fact seeking questions stated in a natural language form*. The proportion of correct answers among the candidate answers must be at least higher than the one dictated by choosing words at random. The technology behind this type of service can be as complex as NLP or as simple as shallow pattern matching. From the designer's perspective, little is known about each service's implementation, so it is treated as a blackbox. We define a meta fact seeking engine as the system that can combine, analyze, and represent the answers that are obtained from several fact seeking services. We call the process of combining, analyzing, and representing the answers as the *recombination mechanism*.

Although keyword based meta search engines have been suggested and explored in the past (e.g. Metacrawler (Selberg and Etzioni, 1995)), we are not aware of a similar approach tried for the task of fact seeking, which we pursue in this paper. We also believe that our proposed approach is a more effective solution in the problem space due to the following important advantages:

1) Eliminating “weakest link” dependency: it does not rely on a single system which may fail or may simply not be designed for a specific type of tasks (questions).

2) Higher coverage and recall of the correct answers since different fact seeking engines may cover different databases or different parts of the Web.

3) Reduced subjectivity by querying several engines; like in the real world, one might need to gather the views from several people in order to make the answers more accurate and objective.

4) The responsiveness provided by several services queried in parallel can also significantly exceed those obtained by working with only one service, since their responsiveness may vary with the task and network traffic conditions. The slower services may be timed out (e.g. as discussed in (Hosanagar, 2005)) to provide a close to real time response.

Challenges Faced and Addressed

Combining multiple fact seeking engines also faces several challenges. First, *the output formats may differ*: some engines produce exact answer (START, NSIR) while others present a sentence or an entire snippet (several sentences) similar to traditional Web search engines (BrainBoost, ASUQA). Figures 2-5 with screenshots illustrate the diversity of their output format. Those differences and other capabilities for the popular fact seeking engines are also summarized below in Table 1. Second, *the accuracies of responses may differ* overall and have even higher variability depending on the specific type of a question. And finally, we have to *deal with multiple answers* and, for this reason removing duplicates, near duplicates, or other answer variations is necessary.

Table 1: The Fact Seeking Services Involved and Their Characteristics			
Fact Seeking Service	**Web address**	**Output Format**	**Organization/System**
START	start.csail.mit.edu	Single answer sentence	Research Prototype
AskJeeves	www.ask.com	Up to 200 ordered snippets	Commercial
BrainBoost	www.brainboost.com	Up to 4 snippets	Commercial
ASU QA	qa.wpcarey.asu.edu[3]	Up to 20 ordered sentences	Research Prototype
Wikipedia	en.wikipedia.org	Narrative	Non-profit
Google	google.com	Up to 200 ordered snippets	Commercial
MSN	msn.com	Up to 200 ordered snippets	Commercial
Google+MSN	n/a	Up to 400 snippets	n/a
Meta (complete configuration)	qa.wpcarey.asu.edu	**Precise answer or up to 100 ranked sentences**	**Research Prototype**

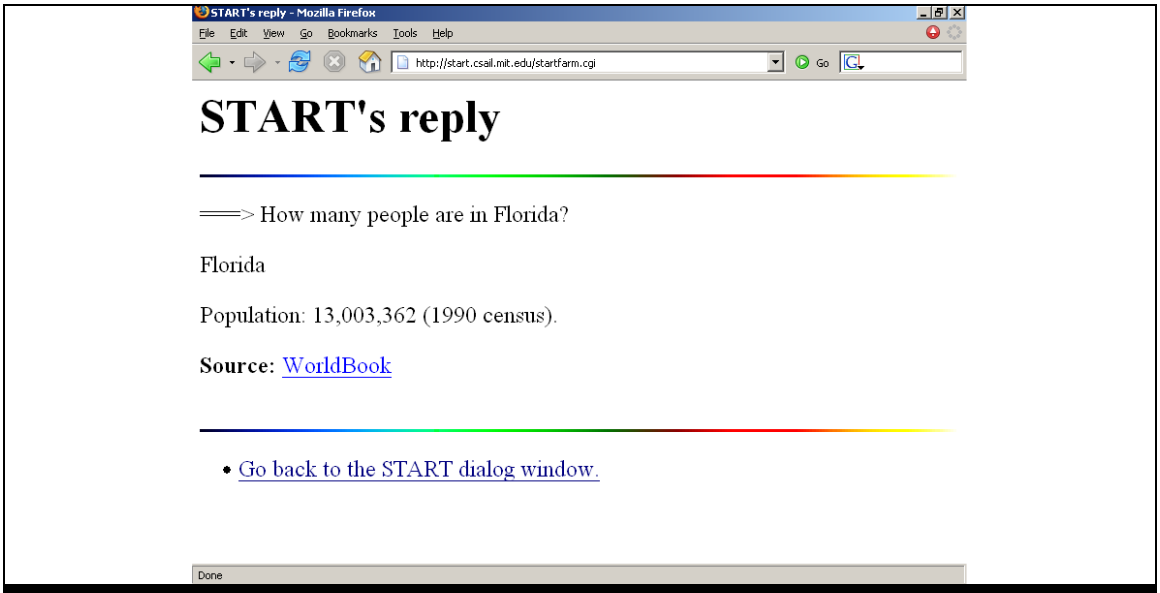

Figure 2. Example of START Output

3 After our study the demo version of ASU QA was changed to the meta engine described in this study.

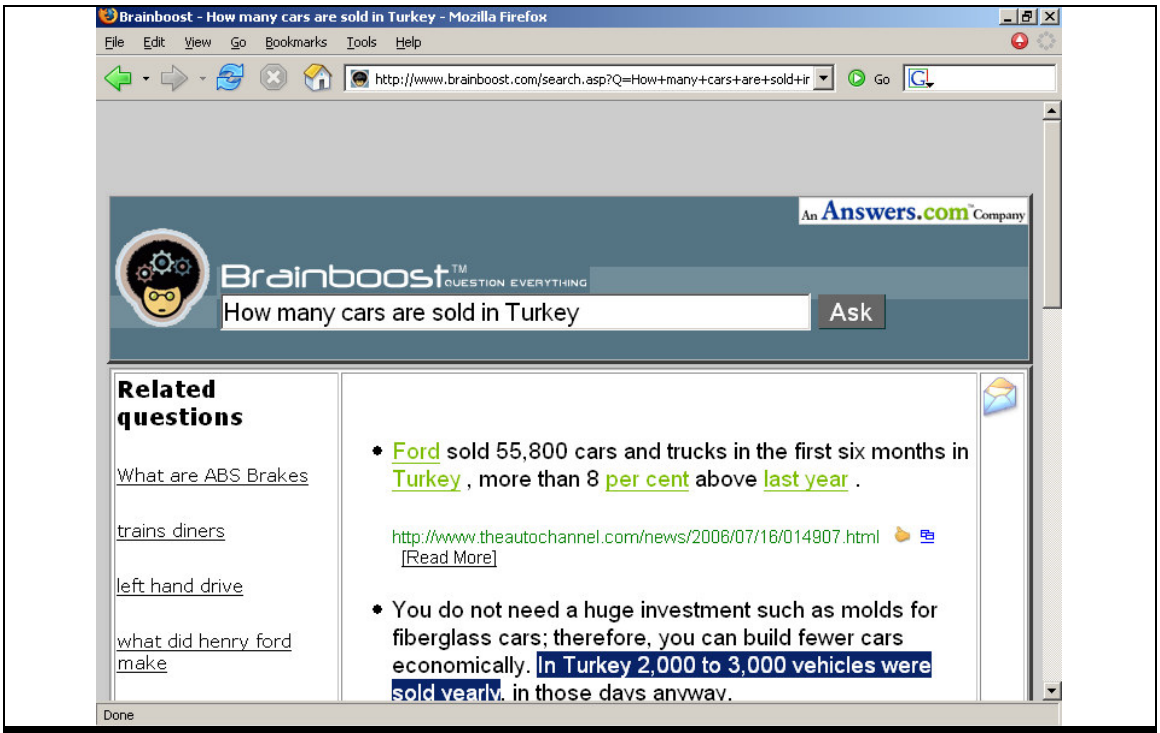

Figure 3. Example of BrainBoost Output

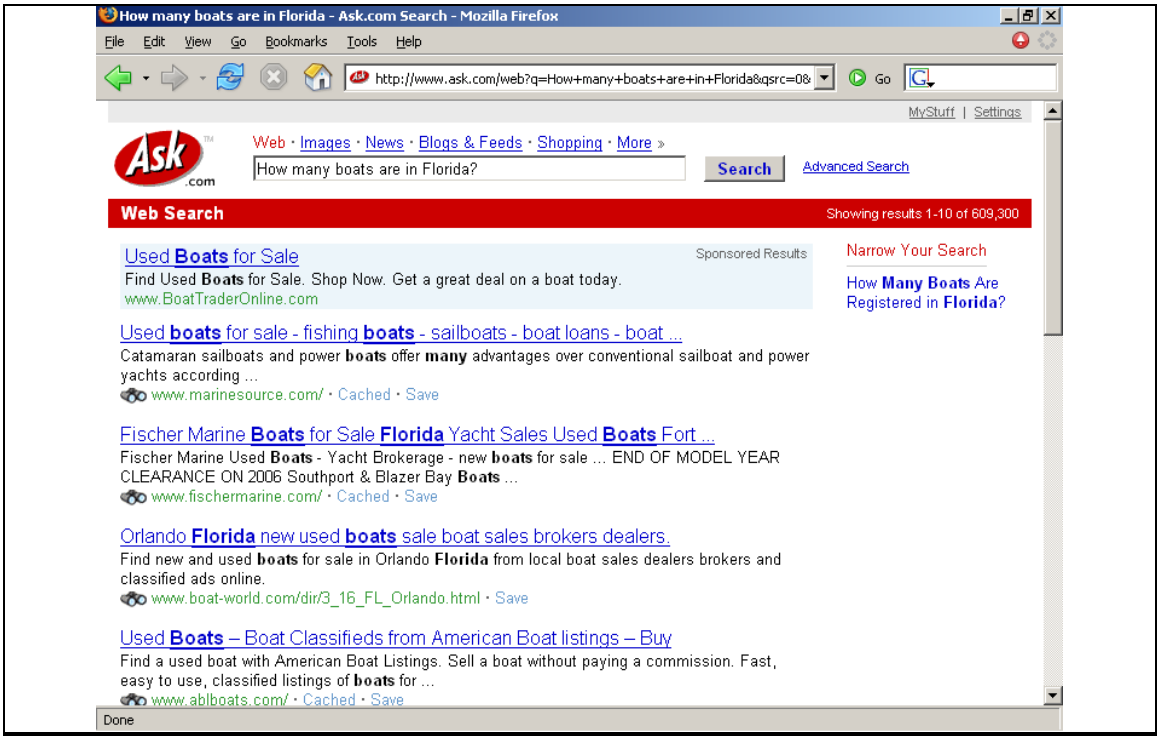

Figure 4. Example of AskJeeves Output

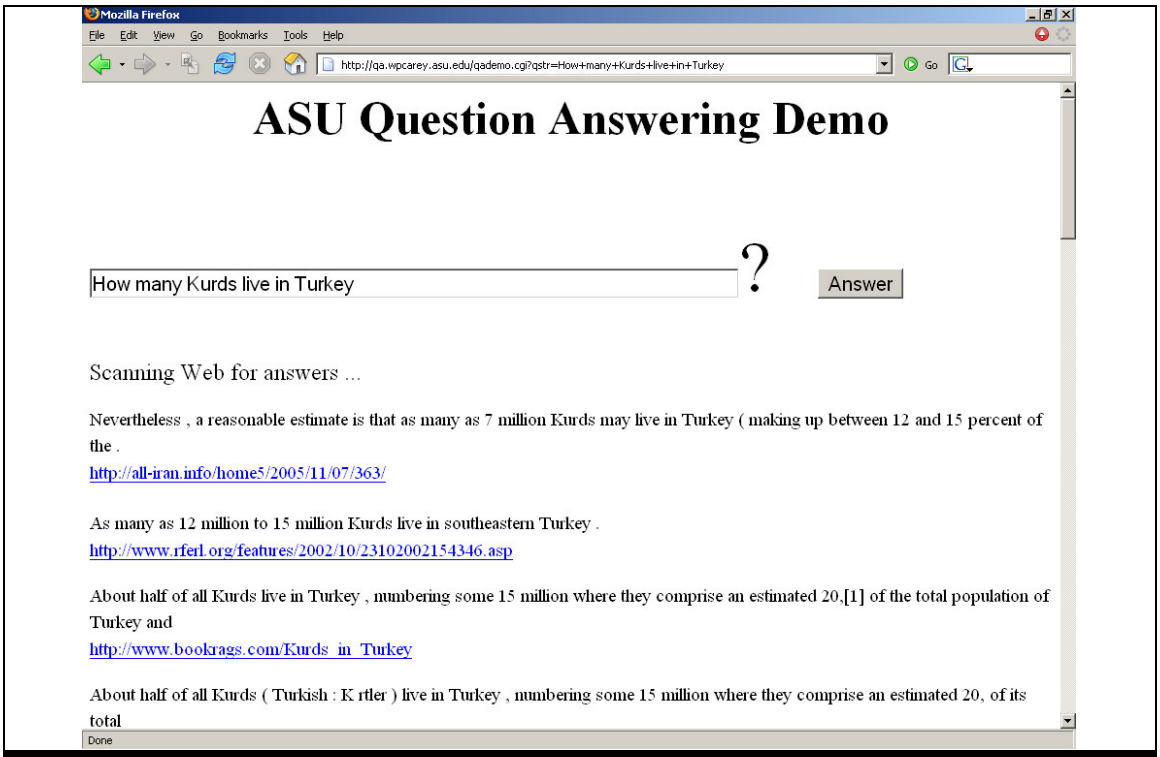

Figure 5. Example of ASU QA Output

The issues with merging search results from multiple keyword-driven engines have already been explored by MetaCrawler (Selberg and Etzioni, 1995), as well as in information fusion studies (e.g., Vogt and Cottrell, 1999) but only in the context of combining ranked lists of retrieved documents. We argue that *the task of fusing multiple answers, which may potentially conflict with or confirm each other, is fundamentally different and poses a new challenge for researchers* which we address here. For example, some answer services may be very precise (e.g. START), but cover only a small proportion of questions. They need to be backed up by a service, maybe a less precise one, that has higher coverage (e.g. AskJeeves). However, backing up may easily result in diluting the answer set by spurious (wrong) answers if the meta engine is not capable of distinguishing right from wrong answers (blind mixing). Thus, *there is a*

need for some kind of triangulation of the candidate answers provided by different services or multiple candidate answers provided by the same service.

Triangulation, a term which is widely used in intelligence and journalism, stands for confirming or disconfirming facts by using multiple sources. In order to employ the full power of triangulation, for each question (e.g. *Who is the CEO of IBM*?), each candidate answer has to be extracted from the sentences returned by answer services (e.g. *Samuel Palmisano* from the sentence *Samuel Palmisano became the twelfth CEO of IBM*), so that the answers can be compared with the other candidate answers (e.g. *Sam Palmisano* -- a possible variation). That is why *the meta engine needs to possess answer understanding capabilities, including such crucial ones as question interpretation and semantic verification of the candidate answers* to check that they belong to a desired category (*person* in the example above).

Research Questions

We would like to emphasize that improving the steps to process a single textual source for question answering task outlined above in the section “Recent Trends in Automated Fact Seeking Technology” (question interpretation, candidate answers identification and assessment, etc.) was not the focus of this study. Rather, *we were primarily interested in exploring whether and when a meta approach to fact seeking offers additional advantages* over approaches studied earlier, such as those of fact seeking engines implemented on top of one or more keyword-driven portals (Agichtein et al., 2001 ; Dumais et al., 2002). We were also interested in the resulting accuracy and responsiveness, in order to evaluate how applicable the meta approach will be in

practice. Inspired by the advantages and challenges discussed in the previous section, we posed the following research questions:

Q1. Is there any value-added of the meta approach: does its performance surpass the performance of each of the contributing services?

Q2. Is it crucial (in terms of performance) to use fact-seeking services as the sources of answers or using keyword-driven search portals is enough?

Q3. Is the approach resilient: how would eliminating one (or several) services impact the overall performance?

Q4. What major components of the answer analysis and triangulation mechanism are crucial when it is applied within a meta framework?

Q5. Does the approach provide practically useful accuracy and responsiveness, especially if contrasted with existing fact seeking services?

To answer our research questions, we have implemented the first, up to our knowledge, prototype of a meta fact seeking engine and performed its empirical evaluation. The technology behind the prototype is explained in the next section.

The Prototype

Overall Architecture of the Prototype

Figure 6 summarizes the overall architecture of the meta approach. Multiple threads are launched to submit the user's question to each fact seeking service and fetch the outputs. In the first version, we have included several freely available demonstrational prototypes and popular commercial engines that have some fact seeking capabilities, specifically START, AskJeeves, BrainBoost, Wikipedia, and ASUQA, as listed in Table

1. We also involved two popular general purpose search portals, namely Google and MSN, in order to answer our research question Q2. Several of those portals are currently providing interface's conforming to the Web Services standards.

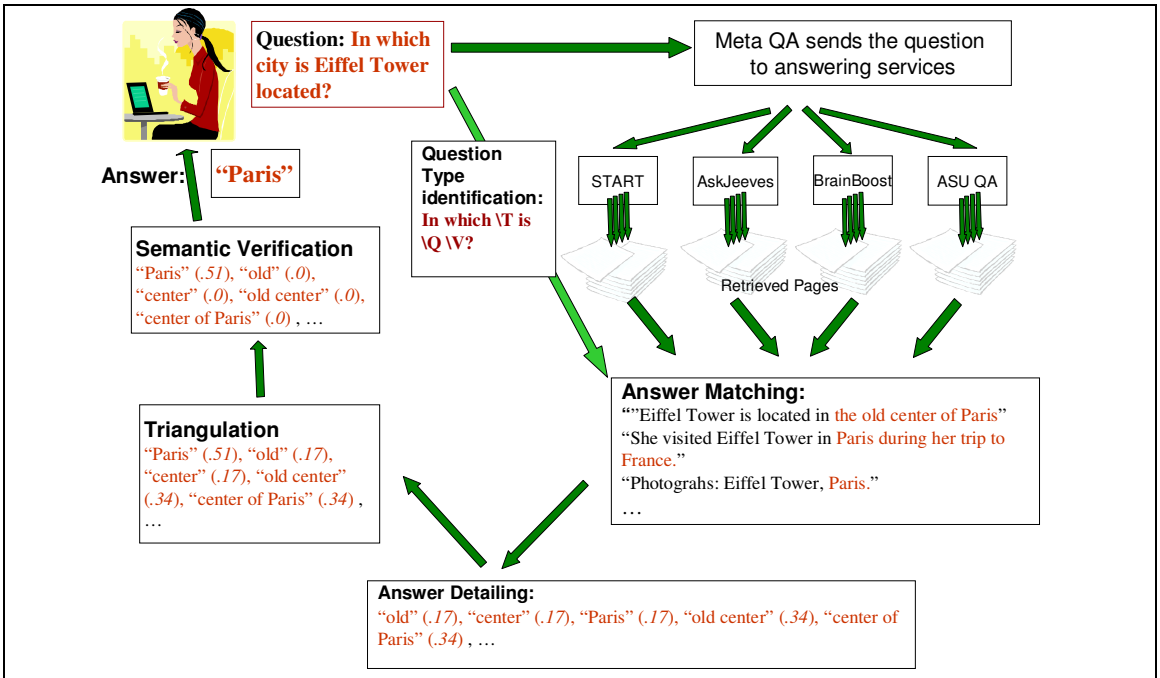

Figure 6. Fact Supply Meta Engine: How it Works

Since none of the services except START produces exact answers, we treat the outputs as sequences of text sentences and apply the answer extraction, triangulation, and semantic verification steps that were applied to a single textual source in prior research (Roussinov et al., 2004). The current prototype is publicly available through a Web interface (http:// qa.wpcarey.asu.edu).

Processing Candidate Answers: Reusing Prior Approaches

This section summarizes briefly the technology that we used to process the outputs from the fact seeking services. It is based on probabilistic pattern matching and triangulation

suggested earlier by several researchers (Clarke et al., 2001; Dumais et al., 2002; Ravichandran and Hovy, 2002; Roussinov and Robles, 2007) to process the outputs from general purpose search engines (e.g. Google). Although the implementation details varied, all their approaches took their roots in the *redundancy* of the Web and automated learning (or manual construction) of the answer *patterns*. The redundancy phenomenon provides that the correct answers are more frequently mentioned in the context of the words contained in the question than are the wrong answers. Although many variations of pattern language have been proposed, they are all essentially trying to capture the possible formulations of answers. For example, an answer to the question *"What is the capital of China?"* can be found in a sentence *"The capital of China is Beijing.",* which matches a pattern *\Q is \A,* where *\Q* is the target of the question (*"The capital of China"*) and *\A = "Beijing"* is the text that forms a *candidate answer. \A*, *\Q*, *\T*, *\p* (punctuation mark), *\s* (sentence beginning), \V (verb) and * (a wildcard that matches any words) are the only special symbols used in our pattern language. *\T* stands for optional semantic category of the expected answer, e.g. for the question *"In which city is Eiffel Tower located?" \T = "city."* In the standard tests, the performances of most of the redundancy/pattern-matching based systems have been found comparable to each other (Voorhees and Buckland, 2006). Their strengths/weaknesses with respect to specific question types have been also found to be similar. For this reason, we believe that our approach used here exemplifies a generic and promising family of approaches, to which our results can be generalized.

Answering the question *"In which city is Eiffel Tower located?"* informally demonstrates the steps of the approach.

Type identification: The question itself matches the pattern *in which \T is \Q located?*, where *\T = "city"*, *\Q = "Eiffel Tower."*

Pattern modulation: It converts each answer pattern into a query for a keyword-driven search engine (KDSE), by replacing \Q and \V with their actual values if they exist. The sequences of words that are not separated by wildcards ('*') or punctuation marks (\p) are surrounded by quotes as KDSE syntax requires for that sequence to be included verbatim in each of the returned pages. For example, the pattern "*\Q is located in \A \p*" would be converted into *"Eiffel Tower" "is located in"*. This heuristic mechanism maximizes the likelihood of the scanned pages to match the answer patterns for a particular question type identified after the previous step.

Answer Matching: The sentence *"Eiffel Tower is located in the old center of Paris"* would match a pattern *\Q is located in \A* and create a candidate answer *"the old center of Paris"* with the corresponding probability of containing a correct answer obtained previously by training on existing questions and known correct answers to them. The training algorithm is summarized below. Its details were first presented in Roussinov et al. (2004). The modulation and matching steps are repeated for each pattern used until the maximum number of candidate answers is reached (1000 in our tests described below). Only the match with the maximum score is extracted from one sentence to avoid double-counting of possibly overlapping candidate answers or those matching several patterns. This provides a closer approximate to the probability of being correct as further elaborated below in the "**Re-ranking Output Sentences**" section.

Answer Detailing: It produces more candidate answers by forming subphrases from the original candidate answers. The subphrases do not exceed three words (not counting "stop words" such as *a, the, in, on*) and do not cross punctuation marks. Each subphrase candidate answer is assigned the same score as the original candidate answer multiplied by the proportion of the length of the subphrase (measured in words) relative to the original match. In our example that would be "old" (*.17*), "center" (*.17*), "Paris" (*.17*), "old center" (*.34*), and "center of Paris" (*.34*). Since both candidate answers "old" and "Paris" have the same length, they are assigned the same scores, although after the next step (Triangulation) we would expect "Paris" to win over "old."

Triangulation: The candidate answers are triangulated (confirmed or disconfirmed) against each other, and then re-ordered according to their final score $s^t(a)$, which is computed by summation as in Roussinov and Robles (2007):

$$s^t(a) = \sum_{a_i \in O} s(a_i) \cdot sim(a, a_i) \quad (1)$$

where *O* is the set of all original (before detailing) answers and *s(a)* is the original score, *sim(a1, a2)* is the similarity between the candidate answers *a1* and *a2,* same way as it was in Roussinov and Robles (2005). Although, there are many known measures of semantic similarity between words and phrases, for simplicity sake, here we used the *relative overlap* measure defined as following: $sim(a1, a2) = 2*|(a1 \cap a2)| / (|a1| + |a2|)$, where $|(a1 \cap a2)|$ is the number of words that are present in both *a1* and *a2*, and $|a|$ is the number of words that are present in *a*. The measure ranges from *0* to *1*, with *1* corresponding to identical answers, and *0* corresponding to no overlap. Although this approach would not detect a similarity between such words as *Sam* and *Samuel*, it would still detect the similarity between *Sam Palmisano* and *Samuel Palmisano*. More

fine-grained approaches may be applied later, e.g. those based on character n-grams (substrings), ontologies, or mined co-occurrences (Soubbotin and Soubbotin, 2003).

Semantic Verification: Similar to Dumais et al. (2002), the approach explored here used a small set of semantic types of questions and a set of 14 adjustment rules that are applied to the score of each candidate answer depending on certain conditions. Table 2 lists all the semantic types used in our prototype, along with some examples of questions and adjustment rules. The conditions were checked automatically by distinguishing between upper or low case of words, regular expressions or dictionary look-ups. The specific adjustment weights were optimized manually on factoid questions from TREC test questions prior to the year 2003 (approximately 1500 in total) not overlapping with those used in our evaluation study described here. When searching for an answer in a closed corpus of documents (e.g. Aquaint collection used in TREC, but not the entire Web), the redundancy based approaches, including the one used here, look for the answer on the Web first and then "project" the answer: using simple heuristic rules, look for the statement inside the close corpus that supports that answer the most. For this reason, their heuristic rules are actually optimized for the performance on the Web rather than using a close corpus. This allowed us to re-use the rules for our "pure" (not involving projection) tests described here without modifications.

Although the specific set of types, rules and adjustment weights that have been used in the prior research (Clarke et al., 2001; Dumais et al., 2002; Ravichandran and Hovy, 2002; Roussinov and Robles, 2007) varied, the number of them and level of sophistication have been relatively the same. Also, the impact of semantic verification on the performance have been reported to be comparable (within 10-20% range), while

increasing number and complexity of rules beyond that resulted in much smaller improvements (less than 5%) (Clarke et al., 2001), most likely because additional rules applied only to a smaller percentage of test questions. We further assume that our verification process is not biased toward any specific configuration or answer source that we have considered in our experiment, thus it would not affect the answers to our research questions since the latter are tested by relative comparisons of the performances.

Table 2: A List of All Semantic Types of Answers Used in the Prototype

Type	Indicators	Examples	Examples of rules	Number of questions
Numeric	Question starts with *how much, how quick, how often, etc.* \T is one of the following: *number, date, time, year, etc*.	How tall a tsunami wave can be? How many justices are members of International Criminal Court? How often does Hale Bopp comet approach the earth ?	If the answer has to be numeric but the candidate answer is not, discount it by 0.01. If the answer has to be non numeric but the candidate answer is numeric, discount it by 0.1.	72
Place	Question starts with *where* or when \T is one of the following: *city, country*, etc.	Where was Kafka born? Where is AARP headquarters? Where do Rhodes scholars study ?	If the candidate answer is not capitalized, discount it by .1	27
Date	Question starts with *when* or \T is *date, year, etc.*	When was Kafka born? When did Floyd Patterson win his title ? What year was the first Concorde crash ?	If the candidate answer is not numeric, discount it by 0.01	40
Person Name	Question starts with *who*.	Who founded the Black Panthers organization ? Who discovered prions ? Who was Horus father ?	If the candidate answer is not capitalized, discount it by 0.05	27
Other	All the remaining questions	What kind of a particle is a quark?	No rules applied	74

Notes: "Date" is also considered to be "Numeric," thus all rows in the last column do not necessary add up to 200.

Pattern Training: We used the same training mechanism as in Roussinov et al. (2004). The purpose of training is to assign to each pattern the probability that the matching text contains a correct answer. We used the questions and correct answers from prior to 2003 TREC competitions to train our patterns. During training, for each pair (*Question, Answer)*, the system requests the Web pages from the search portal (e.g. Google) that have both the question phrase *Q* and the answer *A*, preferably in proximity. Thus, for Google the requesting queries were composed of the \Q and \A as separate words or phrases, each surrounded by quotes, as Google syntax requires for the word or phrase to be included verbatim in each of the returned pages. Each sentence containing both the *Q* and *A* is converted into a candidate pattern by replacing the question phrase with *Q* symbol and the answer with *A*. Once 200 candidate patterns are identified, each pattern is "generalized" to produce more patterns by combining the following:

1. replacing all possible sequences of words (except *A, \Q*) with wildcards,
2. replacing punctuation with *p*,
3. forming all the substrings that still include the symbols *Q* and *A.*

After generalization, for the top 500 most frequent patterns the probability of matching text including a correct answer is estimated as:

prob(P) = # matches containing correct answers / # total matches

where the matches are sought for within the Web pages returned by the search question modulating the pattern (as detailed below) and looking for the matches in the retrieved documents. The training is stopped after at least 40 matches from different pages have been identified. Although the attempts to formalize the estimation of patterns and candidate answers accuracies within a probabilistic framework exist (Downey et al.,

2005; Whittaker et al., 2005), their suggested models have not been empirically shown to be superior to simple heuristic models as the one used here.

Modifications Introduced to the Meta Approach

Although designing novel algorithms to improve the accuracy of a fact seeking process (search) from a single textual source was not the focus of this study, we had to introduce several straightforward but important modifications into the existing fact seeking algorithms in order to be able to use them as a *recombination mechanism* to integrate the outputs from the existing answer services. Those changes may be considered contributing to the novelty of this work and are detailed below.

Weighting the Outputs: Since the accuracy varies among answering services, we believe that treating them in a different way is beneficial. In the current study and prototype, as a first step in that direction, we involved a simple heuristic algorithm to assign different levels of trust to different services. Since the answer matching step described above already involves assigning a score (probability) to each candidate answer based on the accuracy of the matching pattern, we further fine-grained this score assignment by multiplying it by a weight (level of trust) assigned to each service. The weights varied from 0 to 1 and were manually tuned on a set of questions and answers different from the testing set used in our evaluation described below. Thus, less trusted services provided candidate answers with lower scores. Automated approaches, e.g. those based on optimizing the weights through the use of genetic algorithms can be studied in the future.

Re-ranking Output Sentences: Since as we noticed above, many users prefer to read the answers within the surrounding sentences, the meta engine needs to be able to provide the output as a set of rank ordered sentences. Up to our knowledge, the problem of ranking sentences possibly containing correct answers to a fact seeking task, has not yet been explored. As a first step toward that direction and a contribution to the novelty of our work presented here, we have designed and tested a simple heuristic algorithm that ranks the sentences in a decreasing order of the expected total number of contained correct answers:

$$\text{score(S)} = \sum_{\text{c(i)} \in \text{S}} p(i) \ ,$$

where $p(i)$ is the probability of each candidate answer *c(i)* in the sentence S to be correct, which is approximated by the score of the candidate answer after the semantic verification step described above. The aggregate score does not have to be limited to the [0,1] interval.

The intuition behind this approach is the following. Even if the system is wrong about the exact answer but still guesses reasonably well a subphrase or a super-phrase of the exact answer it is still ranks highly a sentence containing the correct answer. By inspecting the logs we observed that in about 50% of the questions that had a correct answer within top 20 but not as the first one, the top ranked sentence still contained the correct answer. For example, the sentence *Samuel Palmisano became the twelfth CEO of IBM* would receive the score of .9 = .5 + .4 if the candidate answers *Samuel* and *Palmisano* have the scores of *.5* and *.4*, respectively. Thus, even the system did not assign a high score to the candidate answer "*Samuel Palmisano"* it would still rank the above sentence higher than those not containing the correct answer at all.

Estimating the expected number of correct answers in this manner assumes the independence of the candidate answers (if considered as random events) that are contained in the same sentence. To make this assumption more realistic (avoid double-counting), we count only the candidate answer with the highest score from each set of overlapping candidate answers.

The independence of candidate answers is justified when no more than one candidate answer is extracted from one sentence and each sentence can be considered an independent event. Two sentences, even identical ones, can be considered as independent events as long as they are not coming from the same or duplicate pages (or segments of pages). More theoretical justification for that assumption was presented by (Downey et al., 2005). Their work also showed that “noisy-or” model used here to triangulate the candidate answers is less accurate than the “urns-and-balls” model. However, the resulting estimate computation is very complex and was tested in a different from our scenario. For those reasons, we leave trying it within fact seeking for future research.

We detected duplication by computing the word overlap between the text windows enclosing those identical sentences. The window was 3 times larger (if possible) in word length than the sentences compared. By manually inspecting our log files, we observed that this approach provided approximately 1% false negatives and 5% false positives. Please note that the false positives (discarded duplicates) only reduce the amount of data to use as evidence, but do not create any bias in favor of any of the candidate answers.

Empirical Evaluation

Exact Answer Evaluation

Mean reciprocal rank or MRR, the first metric that we used, was computed based on the accuracy of the precise answers produced by our meta engine in the ranked order. MRR metric assigns a score of 1 to the question if the first answer is correct. If only the second answer is correct, the score is 1/2, and the third correct answer results in 1/3, etc. The metric penalizes the system for wrong answers but the penalty is decreasing with the rank of the answer. The mean of those reciprocal ranks across all the test questions (MRR) has been the official metric in several TREC QA competitions and used in a number of prior studies cited (e.g. in Dumais et al., 2002). We tested only the top 20 answers and assigned the score of 0 if the correct answer was not there. We also verified that increasing the number of top answers tested from 20 to 100 resulted in scores changing only for a few questions. Since each change could not exceed 1/21 the impact of those changes on the MRR was negligible.

Sentence Level Evaluation

Apparently, the metric described in the previous paragraph may be sensitive to the specific details of our recombination mechanism explained above. However, we do not believe it is a serious limitation since our mechanism is based on the same steps (pattern matching, answer detailing, triangulation by redundancy, and semantic filtering) as many other non-linguistic systems presented in prior research (Clarke et al., 2001; Dumais et al., 2002; Ravichandran and Hovy, 2002; Roussinov and Robles, 2007), thus comprising a very general category. Our implementation of the recombination process, coming from a prior work, was also found exhibiting similar performance and sensitivity

to different types and levels of complexity of questions as the other "knowledge light" systems. Thus we believe our findings here will generalize to the entire category.

It is still possible that our results reported here may differ if a "knowledge heavy" QA system were used as the recombination mechanism instead. However, as we noted above, none of them is currently available for and has been tested with open corpus (Web) fact seeking. It is entirely feasible that "knowledge heavy" approaches have been overtrained for TREC (or similar) competitions and perform even worse than "knowledge light" approaches with an open corpus (Web).

We also looked at the *sentence-level evaluation*, since it can be performed without any manipulation of the output from the answer services and, thus, provide additional insights into the generalizability of our findings. We computed the same MRR metric, but instead of checking for the correctness of the exact answer we checked (also automatically) whether the sentence contained the correct answer using the same regular expression patterns of the correct answers. This sentence-level evaluation is also justified by the consideration that many users prefer to see the answers in context (within sentences or snippets) rather than stand-alone. Thus, the higher the rank of the first sentence containing the answer, the better the system is. This consideration and the need for the sentence-level evaluation in this study necessitated the second modification discussed in the previous section.

Test Sets

We used all the factoid questions from the entire set of questions used by TREC 2004. Table 3 shows more numerical details about our test set. The correct answers found by

all the participants were merged and represented by regular expressions (Voorhees and Buckland, 2006). Examples of questions and their answers are listed in Table 4. We chose the 2004 set because it was the most recent one made publicly available by NIST at the time of our study.

Table 3: A Summary of Our Test Set

Year of the TREC conference	Number of factoid questions	Size of the TREC text collection (Approximately)	Number of documents in the collection (Approximately)
2004	200	1GB	1 Million

Table 4: Examples of Test Questions, Answers Sentences and Precise ("standalone" or "extracted") Answers

Question	Answer Sentence	Precise Answer
Who is the sponsor of International Criminal Court?	United States intends to pull out of the United Nations Criminal Court or the International Criminal Court	United Nations
Where is Rohm and Haas located?	Location : Rohm and Haas Electronic Materials, Blacksburg, VA	Blacksburg, VA
Where is Muslim Brotherhood located?	Most of the violence was reported in Muslim Brotherhood strongholds in the Nile Delta , north of Cairo	Cairo
When was Public Citizen formed?	Public Citizen Formed by Ralph Nader in 1971 to support the work of citizen advocates.	1971
Who is the CEO of the publishing company Conde Nast?	David Carey has been named President of the new business group , announced Charles Townsend, President and CEO of Conde Nast	Charles Townsend
When was the first burger king opened?	Burger King's first restaurant originally called Insta Burger King was opened on December 4, 1954 in Miami , Florida , USA by James.	December 4, 1954
What Las Vegas hotel was made famous by the Rat Pack?	The Rat Pack Live from Las Vegas recreates one of their famous concerts at The Sands, the swinging trio's favorite venue.	Sands
What is the traditional dish served at Wimbledon?	Strawberries and Cream also known as the traditional dish served at Wimbledon.	Strawberries and Cream

NIST and TREC organizers do not have a formal methodology to create test questions, thus their levels of difficulty and distributions by different types vary from year to year. The verbal explanation by NIST during their presentations at the conference briefly

described the procedure as following. Several (5-15) human "authors" were recruited for the process. They were given the instructions on what level of complexity of questions to target. Also, the earlier TREC competitions used the questions from the Excite search engine search logs made publicly available. In the recent competitions, Excite questions were only provided to the "authors" as examples (or "inspiration"). They also had access to the Aquaint collection (Voorhees and Buckland, 2006) of roughly one million documents that was used by recent TREC-s. The same authors of the questions were also assessing the submitted answers to their questions for the correctness.

Although our evaluation has been performed without involving a human user (through a batch mode simulation), we believed that before evaluating at a higher level of cognitive tasks (e.g. decision making), it was first necessary to make sure that the meta approach provides better accuracy at the level of individual questions. We consider our simulation experiment as the first step towards an empirical evaluation involving human participants, which we mention in the concluding section.

Table 5 shows the results of the evaluation of the meta system in several configurations. The last row shows the complete configuration (all sources included). The second column shows the performance (as measured by MRR) when only the service listed on the corresponding row was included. Since all the differences from the complete configuration are statistically significant at the level of .05, the results support our conjecture that using multiple fact seeking services combined through a single meta approach provides more precise answers than each single service does. The third column reports the 95% confidence interval of the relative decrease from the complete configuration. The difference in MRR can be interpreted intuitively in the following way:

Say for example it changes from .3 to .5. It means that typically the correct answer is the second one rather than the third. We believe the differences reported here are practically significant in the light of our motivations (e.g. small screen or time crunch) outlined in the Introduction section. Thus, the answer to our research question Q1 is likely to be positive.

Table 5: The Results of the Tests at the Precise Answer Level

Fact Seeking Service	**Performance if the only service used**		**Performance when the service was excluded**	
	MRR	**Decrease from the complete configuration (%)**	**MRR**	**Decrease from the complete configuration (%)**
START	0.060***	[71%, 97%]	0.486	[-4%, 9%]
AskJeeves	0.412***	[2%, 21%]	0.476	[-5%, 8%]
BrainBoost	0.424**	[1%, 17%]	0.471	[-6%, 7%]
ASU QA	0.416***	[-1%, 22%]	0.475	[-6%, 7%]
Wikipedia	0.211***	[40%, 65%]	0.482	[-5%, 8%]
Google	0.416***	[0%, 21%]	n/a	n/a
MSN	0.355***	[12%, 34%]	n/a	n/a
Google+MSN	0.432**	[-1%, 16%]	n/a	n/a
Meta (complete configuration)	**0.484**	0%	n/a	0%

Notes: ** and *** indicate 0.05 and 0.01 levels of significance of the difference from the complete configuration accordingly. The "% Decrease from complete" columns show the 95% confidence intervals of the decrease of the performance in %, relatively to the complete configuration.

The second column also indicates that BrainBoost was the best source of answer sentences, since its "solo" performance produced the best results of all the five fact seeking services. START did not perform well since it was able to produce answers (although correct ones) only to 6% of the questions. It is worth emphasizing that with the exception of START, the services only supplied text sentences (or several sentences combined into a "snippet") possibly containing the correct answer to the meta engine (e.g. *On New Year's Eve of 2000, the Eiffel Tower played host to Paris' Millennium*.) They did not explicitly state where the precise (standalone) answer (*Paris* for the sentence above*)* was located within those sentences. It was still the meta engine that

was responsible for extracting a standalone answer for the evaluation (here) or presenting to the user in a real life scenario.

The three rows before the last one show the results when keyword-driven portals were used as answer services: Google alone, MSN alone and their combination accordingly. It can be seen that the performances of those combinations were considerably (and statistically significant at the 0.05 level) less than those of the performance of the complete configuration. The observations above suggest that using only the keyword-driven search portals like Google or MSN results in performance drop: 11% and 23% respectively on average, which testify to the importance of using answer services rather than keyword search portals only. This answers positively our research question Q2.

Since adding MSN as an answer source to the configuration using Google as the only source provided only relatively small (4%) improvement, we believe that involving more than two general purpose search portals would not increase the performance much further, which was also found in prior work (Dumais et al., 2002).

Table 6 shows the results of the ANOVA test performed on the data. In the one-way ANOVA test, we compared the mean level of the performance of the services of 9 groups, namely the 5 individual fact seeking services, the 2 general search portals, the combined Google+MSN setting, and the meta fact seeking service with the complete configuration (all fact seeking sources). The ANOVA test result showed that the means of the 9 groups are significantly different at the level *.0001*. To further analyze the relationships among the different services, we conducted Tukey's post-hoc comparisons

and the results are also presented. As discussed above, the complete configuration is significantly better than each of the other configurations. Among the other services, we can see that START and Wikipedia performed significantly worse than the other services. We observed that these two did not contribute much into the meta engine in our case. START produced answers only to a few questions. Wikipedia is not designed to be a fact seeking service since it treats a user's question as a bag (or merely a sequence) of words and finds a related page only for approximately half of the questions. MSN was better than START and Wikipedia, but worse than the other services, possibly because MSN was not designed for fact seeking service. Google, which is also a general-purpose search service, performed better, which is consistent with the public's perception of its being the best search technology at present.

Table 6(a): Results of One-way ANOVA Test (Precise Answer Evaluation)

ANOVA (repeated measures)

Source of Variation	SS	Df	MS	F	*p*-value
Between Groups	28.4886	8	3.5611	43.7737	<0.0001
Within Groups	324.6081	1800			
- error	130.1630	1600	0.0814		
- subjects	194.4451	200			
Total	353.0967	1808			

Table 6(b): *p*-values of Post-hoc Tests (Precise Answer Evaluation)

***p*-values**	**Wiki-pedia**	**Start**	**Brain-Boost**	**Ask-Jeeves**	**ASU**	**MSN**	**Google**	**Google +MSN**
Complete	<0.001	<0.001	0.016	0.003	0.006	<0.001	0.006	0.044
Wikipedia		<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001
Start			<0.001	<0.001	<0.001	<0.001	<0.001	<0.001
BrainBoost				0.553	0.707	<0.001	0.684	0.669
AskJeeves					0.828	0.005	0.853	0.308
ASU						0.003	0.975	0.422
MSN							0.003	<0.001
Google								0.404

The two “Performance when the service was excluded” columns in Table 5 show the performance when each of the services listed on the corresponding row was excluded from the complete configuration. The differences are not statistically significant at the level of *.1*. Since our research question Q3 was stated in the terms of resilience of the meta engine (being not sensitive to excluding one or more of the services) we provide the confidence intervals for each combination. The label “n/a” (not applicable) highlights the fact that our meta engine did not use Google or MSN as fact seeking services in the complete configuration as described above, thus it was not possible to “exclude” them. Google and MSN portals were only used to answer our research question Q2 as described in the previous paragraph.

The results demonstrate *the desired resilience of the meta engine* (positive answer to our research question Q3): the drop in performance even when the best service (BrainBoost) was excluded was relatively small (2.7%). By comparing the 95% interval of the differences in the means (-6.0% to 7.0%) we can see that the relative difference could not exceed 7% with 95% probability. Excluding each of the other services was even less detrimental. This differs from the finding in Lin (2005) with using general purpose search portals for a fact seeking system: excluding one portal resulted typically in 20-30% decrease in accuracy. The different behavior only strengthens our claim that implementing fact seeking engine on top of one (or several) keyword driven search portals is a different task from what we consider here: fact supplying information chain built on public fact seeking services.

The results listed in Table 7 obtained from sentence level evaluation corroborate with the conclusions that we have made above. The “**MRR direct**” column shows the direct score

of each answering service when the meta engine was not transforming its output in any way. It may be intuitive to expect that MRR at the sentence level (Table 7) for the same configuration (choice) of answer services would be higher than MRR measured at precise answer level (Table 5), since sentence level evaluation is more lenient: at the sentence level it is enough for the correct answer to be included in the sentence to be credited as a correct one for the reciprocal rank computation, while at the answer level, the candidate answer should match one of the correct answers exactly (verbatim, please see Table 4 for the examples to clarify the difference). That is why it is important to clarify that we observed that this inequality did not always hold. The following example offers an explanation. We observed several cases where the first answer was wrong, and assigned the score (approximation of the probability of being correct) 10 times (or more) higher than the second answer, which happened to be correct. Thus, the MRR at the exact level was 1/2. However, the first (and erroneous) answer happened to be present in a large number of sentences and, as a result, many of them were ranked highly and taking top 9 positions. In that situation, the sentence level MRR could not exceed 1/10, which was much smaller than MRR at the exact answer level.

The results indicate that sentence-level performance varies significantly among services. Again, BrainBoost emerged as the leader, statistically different from all the others at the *.1* level of significance. It is clearly visible that the performance of each service was well below the performance of the meta engine studied here thus reinforcing our positive answer to Q1. All the other results shown in the “MRR direct” and “MRR if the only service used “ columns in Table 7 are statistically different from the performance of the complete configuration (.630) at the level of *.01*. The second last row shows the sentence level performance when only keyword driven search portals were used. It

provides additional evidence for our positive answer to Q2. Again, the label "n/a" (not applicable) indicates that our meta engine did not use Google or MSN as fact seeking services in the complete configuration. Another "n/a" indicates that combination of Google and MSN can not be evaluated directly without involving some kind of answer recombination mechanism.

Table 7: Sentence-level Evaluation of the Individual Services and Their Contributions

Fact Seeking Service	MRR direct	MRR if the only service used	Performance if excluded	
			MRR	Decrease from the complete configuration (%)
START	0.050***	0.050***	0.628	[-3%, 7%]
AskJeeves	0.372***	0.402***	0.622	[-4%, 6%]
BrainBoost	0.422***	0.433***	0.610	[-3%, 5%]
ASU QA	0.314***	0.367***	0.635	[-7%, 4%]
Wikipedia	0.274***	0.302***	0.626	[-3%, 6%]
Google	0.251***	0.344***	n/a	
MSN	0.214***	0.305***		
Google +MSN	n/a	0.425***		
Meta	**n/a**	**0.630**		

Notes: In the second and third columns, *** indicates 0.01 level of significance of the difference from the complete configuration. The final column shows the 95% confidence interval for the decrease of the performance relative to the complete configuration.

The "MRR if the only service used" column presents the performance when each individual service was the only source of candidate answers, while the meta engine was still performing triangulation and semantic verification. It is interesting to note that the data suggests that all of the individual services (except START and BrainBoost) can possibly improve their performance (at least as measured by MRR on TREC questions) if they apply the same redundancy-based triangulation algorithm that we have involved in this study. One reason that they have not accomplished it yet is that some engines, like Google, MSN, and Wikipedia, are not designed to be fact seeking services. As we

noted above, they treat a user's question as a bag or a sequence of words. AskJeeves also most often resorts to keyword interpretation of a user's question.

The last two columns illustrate the resilience of the meta approach at the sentence level by presenting the MRR of the system when the service on the corresponding row was excluded and the 95% confidence intervals of the relative changes. The results again support a positive answer to Q3: when each of the services was excluded none of the changes was practically significant.

Table 8(a): Results of One-way ANOVA Test (Sentence-level Evaluation)

ANOVA (repeated measures)

Source of Variation	SS	Df	MS	F	*p*-value
Between Groups	33.6258	8	4.2032	34.7776	<0.0001
Within Groups	308.6690	1800			
- error	193.3762	1600	0.1209		
- subjects	115.2927	200			
Total	342.2948	1808			

Table 8(b): *p*-values of Post-hoc Tests (Sentence-level Evaluation)

***p*-values**	**Wiki-pedia**	**Start**	**Brain-Boost**	**Ask-Jeeves**	**ASU**	**MSN**	**Google**	**Google +MSN**
Complete	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001
Wikipedia		<0.001	<0.001	<0.001	0.008	0.904	0.084	<0.001
Start			<0.001	<0.001	<0.001	<0.001	<0.001	<0.001
BrainBoost				0.204	0.007	<0.001	<0.001	0.989
AskJeeves					0.155	<0.001	0.020	0.209
ASU						0.012	0.364	0.007
MSN							0.108	<0.001
Google								<0.001

In addition, we also performed one-way ANOVA test on the sentence-level evaluation data (corresponding to the third column in Table 7). Again, the results, presented in

Table 8, showed that the mean performances of the 9 settings are significantly different. Tukey's post-hoc tests results are also shown in the table. Besides that the complete configuration performed the best, we can see that the remaining services can be roughly classified into 3 groups, namely, BrainBoost, AskJeeves, and Google+MSN on the high end, ASU, Google, MSN, and Wikipedia in the middle, and START on the low end.

Although the impacts of the major steps within the redundancy based approach to question answering has been explored before (Clarke et al., 2001), we desired to verify them in our case of meta fact seeking. In order to test what components of the meta engine were essential (Q4), we run the tests with some of the components disabled and computed MRR at the exact answer level. The results are shown in Table 9. All the pair-wise differences were statistically significant at the level of *.1* (t-tests) except between "same weights" and "complete". When the patterns were not used while looking for the answers among the results returned by the underlying services, the meta engine relied only on the redundancy (looking for the most repeated substring) and on verifying the expected semantic category of the answer (person, place, etc.). The performance dropped only 11%, illustrating previously known observations (e.g. Clarke et al., 2001) that the redundancy (repetitions) is a powerful indicator of correctness, and that in general using the grammatical patterns in addition to the redundancy does not contribute that much as someone would intuitively expect. When no semantic verification was performed, the performance dropped more, which shows that semantics plays a very important role in fact seeking, maybe even more important than the grammar captured by the answers patterns. When no pattern was used and no semantic verification was applied, the meta engine relied solely on redundancy and did not need to understand the question at all: it blindly looked for the phrases most repeated in the outputs from the

services. However, the performance was very low in that case. Those observations clearly illustrate that *the meta engine needs to possess question understanding capabilities and can not just blindly combine results of the underlying services.* This is a fundamental difference from the meta approach applied to keyword-based retrieval (Selberg and Etzioni, 1995) where *simple linear re-combination of the relevance scores of the retrieved results always resulted in comparable accuracy*.

Table 9: The Performance of the Reduced Configuration of the Meta Fact Seeking Engine

Configuration	MRR
Complete	0.484
Same weights	0.442*
No patterns	0.430*
No semantic verification	0.397**
No patterns, and no semantic verification	0.354**

Notes: The results of the tests at the precise answer level. * and ** indicate 0.1 and 0.05 levels of significance of the difference from the complete configuration accordingly.

By analyzing the processing logs along with the time stamps, we observed that on average 75% of services replied within the 25% interval of the longest wait time. We estimated that by allowing the system to time-out (stop waiting for) the slowest service in each request, the total wait time could be cut by approximately 50%. If we allow 2 services to time-out, then the total wait time can be cut 70% and become 2.5 seconds in average. Due to the observed resilience (Q3) one or two slowest services can be timed out without much loss in the accuracy. Thus, we conclude that the meta approach provides responsiveness superior to each individual service and can be used within practical applications, which is currently unfortunately not the case with standalone answering services due to their occasionally slower responses or lower accuracy. This, we believe, answers our research question Q5.

We also ran similar tests a year before this study. While the absolute values of the measurements were slightly (no more than 10%) different, the relative differences were consistent with the findings reported here. We believe this not only makes our claims stronger but also indicates longitudinal independence of the findings relatively to the state of the Web. We admit that in several years and after more technological breakthroughs the conclusions provided here may need to be modified.

Conclusions, Limitations and Future Research

Following the design science principles (Hevner et al., 2004), we have suggested and evaluated a prototype called *meta fact seeking engine*. It combines several other independent online fact seeking (question answering) services within a single information supply chain. Even if each of the combined services may not be 1) accurate, 2) comprehensive, 3) responsive, or 4) reliable, the recombination mechanism, taken from prior research and adapted for the meta engine application as described above, results in a chain that is improved along all those four dimensions.

We performed a batch mode evaluation with the currently available question answering services and established the following: 1) Value-added of the meta approach: its performance surpassed the performance of each contributing service. 2) The importance of using fact seeking services rather than general purpose search portals (Google and MSN). 3) The resilience of the accuracy of the combination to exclusion (e.g., timing out) each individual service. We further conclude that the overall performance of the prototype as measured by the responsiveness and accuracy is sufficient to be applicable in practical every-day tasks, which is in contrary to the currently offered fact seeking services on the Web if used in isolation. Indeed, the sentence level evaluation (MRR of

.630) implies that on average the correct answer is contained within the first or second output sentence, while each service separately provided MRR under .433, which places correct answer typically in the second or third sentence. The estimated 70% cut in the response time down to 2.5 seconds in average provides necessary responsiveness, which current services are missing.

The managerial implications of our findings are that:

1) If properly designed and implemented, fact seeking technology can be practically useful for business intelligence and monitoring, especially when having precise answers is extremely desirable, e.g. while using mobile devices, voice interfaces, time crucial application or systems for visually impaired people.

2) A meta approach seems to be a better approach than relying on each individual fact seeking service, at least at the current level of technology. By combining information services provided by different information suppliers, it is possible to provide better and richer services.

Although our findings are somewhat dependent on the specific recombination technology that we used and the heuristics embedded in it, we believe this limitation is not serious since the technology falls into a generic and becoming popular category of "knowledge-light" redundancy-based fact seeking approaches, with all currently known instantiations demonstrating similar performance and behavior (e.g. dependence of the accuracy on the question type). More detailed exploration on finding the minimum set of heuristics and the possibility of automatically discovering them may be studied in future.

No major resources are necessary to implement a meta fact seeking engine. Its set of manually tuned heuristic rules is small. It uses very few linguistic resources: namely part of speech tagging, list of common words along with their part of speech, list of all countries, US cities and states spelled in various forms (e.g. VT, Vermont, Verm.), list of all words that may constitute a number and list of the most common measurement units (foot, meter, hour, etc.). All of those resources are publicly available or can be downloaded from our Web site (http://qa.wpcarey.asu.edu) along with the current set of answer patterns, which could be independently trained using the algorithms described here and in prior work. The processing is not computationally expensive. For illustration, we notice that most of our tests were run on Dell Latitude D620 laptop in background, without interrupting or slowing down the laptop user.

The current bottleneck for the overall speed is waiting to hear from the contributing services. Waiting and processing their outputs is currently taking between 5 and 12 seconds. However, the current implementation emphasized simplicity and transparency of the code in order to be able to run potentially replicable tests. It has not been optimized for speed. Since none of the steps of the algorithm is really time consuming (e.g. requiring iterating large lists, intensive reading from the hard drive or high-order or nonpolynomial complexity), we are certain that the processing time can be reduced to that being negligible relatively to the response times from the services. Thus, the meta systems that have sufficiently fast access (e.g. through T3 or LAN lines) to the services may achieve the response under a fraction of a second. Large corporations or various government agencies can negotiate or sometime already have that kind of access to the major Internet portals.

Our suggested algorithmic modifications in order for the meta approach to be applicable are straightforward and heuristic in this study, which may limit somewhat their performance and the generalizability of the findings. Thus, we are leaving exploring more theory driven approaches to future research. For example, future implementations may automatically learn the accuracy of each service with respect to a specific question type and apply the learned weights discriminatively.

We are not addressing any possible issues that may arise from using commercial fact seeking services and thus possibly "stealing" their advertising revenue. As our results indicate, there are enough non-commercial services (research prototypes) at present to provide good performance. Advertising revenue sharing models may be considered in future if meta supply chains were to become popular portals. For example, the source may automatically receive a credit when the user clicks on the answer provided by that source.

Evaluation not involving a user, through a batch mode simulation, has its limitations too, which we are currently overcoming through a controlled experiment. Nevertheless, we believed that before going to higher level cognitive tasks (e.g. decision making) it was necessary to test the improvement provided by the meta approach through a "batch mode" simulation at the level of individual fact seeking tasks (questions). Another future direction will be field-testing our prototype within a specific organization.

Acknowledgments

The authors would like to thank Kevin Lee and Boby Shiu at the University of Hong Kong for their programming help in this project. We also thank TREC for making their test data available for research.

References

1. Agichtein, E., Lawrence, S., Gravano, L. (2001). Learning Search Engine Specific Query Transformations for Question Answering, In Proceedings of the Tenth International WWW Conference (10), Hong-Kong.
2. Castellani, F. (2004). Program Cracks Crosswords. *Nature*, 10/04/04.
3. Chen, H., Chau, M., and Zeng, D. (2002). "CI Spider: A Tool for Competitive Intelligence on the Web," Decision Support Systems (DSS), 34(1), 1-17, 2002.
4. Chung, W., Chen, H., and Nunamaker, J. (2005). A Visual Knowledge Map Framework for the Discovery of Business Intelligence on the Web, *Journal of Management Information Systems*, (21:4), Spr 2005, pp 57-84.
5. Clarke, C., Gormack, G., and Lyman, T. (2001). Exploiting redundancy in question answering. In proceedings of the 24th Annual International ACM SIGIR Conference on Research and development in information retrieval, pp. 358-365.
6. Downey, D., Etzioni, O., and Soderland S. (2005). A Probabilistic Model of Redundancy in Information Extraction, Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI 2005).
7. Dumais, S., Banko, M., Brill, E., Lin, J., and Ng, A. (2002). Web Question Answering: Is More Always Better? Proceedings of the 25th annual international

ACM SIGIR conference on Research and development in information retrieval, Tampere, Finland, August 11-15.

8. Harabagiu, S., Moldovan, D., Pasca, M., Mihalcea, R., Sudenu, M., Bunescu, R., Girju, R., Rus, V., and Morarescu, P. (2000). Falcon: Boosting knowledge for answer engines. In NIST Special Publication 500-249: The Ninth Text REtrieval Conference (TREC 9), pages 479–488, Gaithersburg, Mary-land, November 13-16.
9. Hevner, A.R., March, S.T., Park, J., Ram, S. (2004). Design Science in Information Systems Research. *Management Information Systems Quarterly*, 28 (1), pp. 75-105, March 2004.
10. Hosanagar, K. (2005). A Utility Theoretic Approach to Determining Optimal Wait Times in Distributed Information Retrieval. SIGIR 2005.
11. Katz, B. (1997). From Sentence Processing to Information Access on the World Wide Web. In Natural Language Processing for the World Wide Web: Papers from the 1997 AAAI Spring Symposium, pages 77-94, 1997.
12. Katz, B., Bilotti, M., Felshin, S., Fernandes, A., Hildebrandt, W., Katzir, R., Lin, J., Loreto, D., Marton, G., Mora, F., Uzuner, O. Answering Multiple Questions On a Topic From Heterogeneous Resources. Proceedings of the Thirteenth Text REtrieval Conference (TREC 2004), November 2004, Gaithersburg, Maryland.
13. Kwok, Cody and Etzioni, Oren and Weld, Daniel S. (2001) Scaling Question Answering to the Web. In Proceedings of the Tenth International WWW Conference(10), Hong-Kong.
14. Lam, S., Pennock, D., Cosley, D., Lawrence, S. (2003). 1 Billion Pages = 1 Million Dollars? Mining the Web to Play Who Wants to be a Millionaire? Proceedings of the 19th Annual Conference on Uncertainty in Artificial Intelligence (UAI-03), pp. 337-345, 2003.

15. Lin, J. (2005). Evaluation of Resources for Question Answering Evaluation. Proceedings of ACM Conference on Research and development in information retrieval, 2005.
16. Marcus, M. (1995). New Trends in Natural Language Processing: Statistical Natural Language Processing, *Proceedings of the National Academy of Sciences*, Vol 92, 10052-10059.
17. McGonagle, J. J. and Vella, C. M. (1999). The Internet Age of Competitive Intelligence, London: Quorum Books 1999.
18. Radev, D. R., Libner, K., and Fan, W. (2001). Getting answers to natural language queries on the Web. Journal of the American Society for Information Science and Technology (JASIST), 53(5):359-364.
19. Radev, D., Fan, W., Qi, H., Wu, H., and Grewal, A., (2005). Probabilistic question answering on the Web, Journal of the American Society for Information Science and Technology, 56(3).
20. Ravichandran, D., and Hovy, E. (2002). Learning surface text patterns for a question answering system. In Proceedings of ACL, 2002.
21. Roussinov, D. and Robles, J., Ding, Y. (2004). Experiments with Web QA System and TREC2004 Questions. In the proceedings of TREC conference. November 16-19, 2004, Gaithersburg, MD.
22. Roussinov, D., and Robles-Flores, J.A. (2005). Web Question Answering: Technology and Applications to Business Intelligence, *International Journal of Internet and Enterprise Management* (3:1) 2005, pp 46 - 62.
23. Roussinov, D., and Robles-Flores, J.A. (2007). Applying Question Answering Technology to Locating Malevolent Online Content. *Decision Support Systems,* 43 (4), 2007, pp. 1404-1418.

24. Selberg, E. and Etzioni, O. (1995). Multi-Service Search and Comparison using the MetaCrawler. Proceedings of the 4th World Wide Web Conference, Boston, MA, USA, December 1995.
25. Soubbotin, M. and Soubbotin, S. (2002). Use of patterns for detection of likely answer strings: A systematic approach. Proceedings of the Eleventh Text Retrieval Conference TREC 2002. Gaithersburg, Maryland, November 19-22.
26. The Economist. (2005). Internet advertising. April 27th, 2005 issue.
27. Vogt, C. and Cottrell, G. (1999). Fusion Via a Linear Combination of Scores. Information Retrieval, 1(3), pp. 151-173.
28. Voorhees, E. (2003). Overview of TREC 2003. Proceedings of the Text Retrieval Conference TREC 2003, Gaithersburg, Maryland, USA.
29. Voorhees, E. and Buckland, L.P. (2006). Proceedings of the Fifteenth Text Retrieval Conference TREC 2006. Gaithersburg, Maryland, November 16-19, 2006.
30. Whittaker, E., Chatain, P., Furui, S., Klakow, D. (2005). TREC2005 Question Answering Experiments at Tokyo Institute of Technology, In NIST Special Publication 500-249: The 2005 Text REtrieval Conference (TREC 9), pp. 479–488, Gaithersburg, Maryland, November 13-16.

About the Authors

Dmitri Roussinov is an Assistant Professor in the Department of Information Systems, W.P. Carey School of Business, Arizona State University. He received his Ph.D. in MIS from the University of Arizona and has a prior MA degree in Economics from Indiana University, and a diploma with honors in Computer Science from Moscow Institute of Physics and Technology, Russia. Prior to joining ASU, Dr. Roussinov served two years

on the faculty at Syracuse University, School of Information Studies. His research interests include security informatics, applications of artificial intelligence to knowledge management, group decisions support systems, and electronic commerce. Dr. Roussinov has published in *IEEE Transactions on Knowledge and Data Engineering*, *Communications of the ACM*, *Decision Support Systems*, and *Information Processing and Management*. He has also presented at many well known international conferences.

Michael Chau is an Assistant Professor in the School of Business at the University of Hong Kong. He received a Ph.D. degree in Management Information Systems from the University of Arizona and a bachelor degree in Computer Science and Information Systems from the University of Hong Kong. His current research interests include information retrieval, Web mining, data mining, knowledge management, electronic commerce, security informatics, and intelligence agents. He has published more than 60 research articles in various journals and conferences including *IEEE Computer*, *ACM Transactions on Information Systems*, *Journal of the America Society for Information Science and Technology*, *Annual Review of Information Science and Technology*, *Decision Support Systems*, *International Journal of Human-Computer Studies*, *Communications of the ACM*, and *International Conference on Information Systems*. More information can be found at http://www.business.hku.hk/~mchau/.