An Exploration of the Principles Underlying Redundancy-Based Factoid Question Answering

JIMMY LIN
University of Maryland, College Park

The so-called "redundancy-based" approach to question answering represents a successful strategy for mining answers to factoid questions such as "Who shot Abraham Lincoln?" from the World Wide Web. Through contrastive and ablation experiments with Aranea, a system that has performed well in several TREC QA evaluations, this work examines the underlying assumptions and principles behind redundancy-based techniques. Specifically, we develop two theses: that stable characteristics of data redundancy allow factoid systems to rely on external "black box" components, and that despite embodying a data-driven approach, redundancy-based methods encode a substantial amount of knowledge in the form of heuristics. Overall, this work attempts to address the broader question of "what really matters" and to provide guidance for future researchers.

Categories and Subject Descriptors: H.3.4 [**Information Storage and Retrieval**]: Systems and Software

General Terms: Design, Experimentation, Measurement, Performance

Additional Key Words and Phrases: Data redundancy, Web search

ACM Reference Format:

Lin, J. 2007. An exploration of the principles underlying redundancy-based factoid question answering. ACM Trans. Inform. Syst. 25, 2, Article 6 (April 2007), 55 pages. DOI = 10.1145/1229179.1229180 http://doi.acm.org/10.1145/1229179.1229180

1. INTRODUCTION

The vast amounts of information available on the World Wide Web make it an attractive resource for answering a variety of user questions. However, its usefulness as a repository of human knowledge is limited by the effectiveness of available means for accessing it. The sheer size of the Web and its lack of central organization often overwhelm casual and professional information

I would like to thank Esther and Kiri for their kind support, under Grant #2006-05-26.
Author's address: College of Information Studies, University of Maryland, College Park, College Park, MD 20742; e-mail: jimmylin@umd.edu

seekers alike. Web search engines frequently return tens if not hundreds of thousands of "potentially relevant" pages in response to a query, leaving users to manually sift through long hit lists.

Question answering (QA), which lies at the intersection of natural language processing and information retrieval, has recently emerged as a technology that promises to deliver "answers" instead of "hits." Much research has focused on fact-based questions such as "When did Montana become a state?" "Who invented the paper clip?" and "How far is it from the pitcher's mound to home plate?" These so-called "factoid" questions can be typically answered by named entities such as dates, locations, people, organizations, measures, etc.

This article describes experiments with Aranea, a system that exemplifies what has come to be known as the redundancy-based approach to factoid question answering using data from the World Wide Web. The system has consistently performed well at TREC question answering tracks [Voorhees 2001, 2002, 2003], a series of annual evaluations organized by the National Institute of Standards and Technologies (NIST). These QA evaluations, which began in 1999, represent a commonly accepted benchmark for measuring system performance.

At the broadest level, this work attempts to tackle the question of "What matters in redundancy-based factoid question answering?" Although the TREC QA tracks undoubtedly play an important role in advancing question answering technology, their end-to-end focus means that it is difficult to untangle the performance contributions of individual components without additional experiments (e.g., Moldovan et al. [2002]). While it is possible to conduct component-level evaluations outside the TREC framework (and indeed many participants do so), reporting of such results is, for the most part, less than systematic. In the same vein, although Aranea has been previously described in a number of papers [Lin et al. 2002; Lin and Katz 2003], no such detailed study has been conducted. Fortunately, the system's modular architecture facilitates component-level and one-off experiments designed to explore the many factors that affect question answering performance. Through a series of ablation studies and contrastive runs, this work attempts to refine our understanding of redundancy-based techniques. Specifically, two theses are articulated:

—The redundancy-based approach to factoid question answering evolved from methods driven by large ontologies of question and answer types. In contrast, redundancy-based techniques are often believed to be "knowledge poor" [Brill et al. 2001]—an assumption that we challenge. Although earlier works touted data redundancy as the primary driver of performance, experiments with Aranea demonstrate that heuristic knowledge also plays an important role in the question answering process.

—Large-scale applications are typically built on individual components with well-known and stable characteristics. This engineering principle allows for hierarchical decomposition of functionality, which makes system-building practical and manageable. However, many redundancy-based systems rely on external "black box" components whose behavior at the input/output interfaces are ill-defined (and worse, constantly changing). Nevertheless,

experiments suggest that the Web exhibits a number of stable properties that appear to be invariant with respect to many confounding variables. This finding allows factoid QA systems to rely on existing Web search engines as a source for candidate answers.

The organization of this article is as follows: first, we frame the context for this work by providing a historical overview of factoid question answering and discussing the property of data redundancy (Section 2). The performance of redundancy-based techniques, as measured by the TREC question answering evaluations, is presented in Section 3. Aranea's architecture and its processing modules are detailed in Section 4. An overview of the experimental methodology occupies Section 5. Experimental results are presented in three separate sections: Section 6 focuses on the behavior of external components, Section 7 focuses on the role of knowledge, and Section 8 focuses on the impact of other system settings. We then attempt to take a broader view of the redundancy-based approach: Section 9 generalizes experimental results and discusses their implications. This article concludes with Section 10.

The entire source code of Aranea was released under an open source license in May 2005, providing the community with a stable code base and a platform for further experimentation. The complete system can be downloaded at `http://www.umiacs.umd.edu/~jimmylin/downloads/`.

2. RESEARCH CONTEXT

Although the roots of question answering can be traced back to the 1960s, modern open-domain question answering can be viewed as a fine-grained retrieval task. For much of the history of information retrieval research, the document has occupied a privileged position as the basic unit of retrieval. However, the recognition that subdocument segments (i.e., passages) may better satisfy user information needs lead to the development of passage retrieval techniques e.g., Salton et al. [1993]; Moffat et al. [1993]; Mittendorf and Schäuble [1994]; Zobel et al. [1995], of which question answering can be seen as the logical limit. This section situates our exploration of redundancy-based techniques with a historical account of its development, followed by a discussion of data redundancy, the key principle underlying this approach.

2.1 Development of the Redundancy-Based Approach

In the late 1990s, the dominant approach to answering open-domain factoid questions involved a combination of information retrieval and named-entity recognition technology. The typical architecture of such a system is shown in Figure 1, but see also Hirschman and Gaizauskas [2001] and Tellex et al. [2003]. These systems are usually driven by large ontologies that relate question types to semantic classes of answers. In the question analysis stage, knowledge resources are leveraged to determine the expected semantic type of the answer. In the document retrieval stage, candidate documents containing terms from the question are retrieved (most often, using off-the-shelf IR engines). Next, the system identifies passages within these candidate documents that contain a concentration of terms from the question—these regions are likely to contain

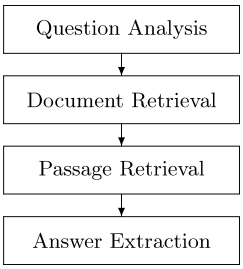

Fig. 1. The architecture of a "traditional" ontology-driven question answering system based primarily on information retrieval and named-entity recognition technologies.

the answer. Finally, in the answer extraction stage, named-entity recognizers identify candidates of the correct semantic type.

The main complexity of this approach lies in the complex many-to-many mapping between question types and answer types. For example, the answer to a "who" question can be person's name (e.g., "Who invented the light bulb?") or the name of an organization (e.g., "Who won the World Series in 2004?"), among other possibilities. Similarly, multiple question types can map onto the same semantic answer type, for example, compare "*Where* was the world fair in 1900?" to "*What city* hosted the world fair in 1900?" To overcome these challenges, systems required the support of elaborate ontological resources that explicitly encoded semantic relationships between questions and answers. Notable systems that adopt this approach include those described by Srihari and Li [1999], Harabagiu et al. [2000a], Hovy et al. [2000], and Prager et al. [2000]; for convenience, we refer to this as the "traditional" approach.

Both the implementation of question answering evaluations at TREC starting in 1999 [Voorhees and Tice 1999]—modeled after the familiar ad hoc retrieval task—and the background of early QA researchers—mostly from the fields of information retrieval and information extraction—contributed to the prevalence of the "traditional" architecture. Due to the empirical success of this conceptually simple strategy at the first few TREC evaluations, it evolved into the standard paradigm for factoid question answering; see Hirschman and Gaizauskas [2001] for an overview. Although researchers have since developed more advanced techniques such as abductive inferencing [Harabagiu et al. 2000b], feedback loops [Moldovan et al. 2002], and fuzzy matching of syntactic relations [Cui et al. 2005], factoid question answering is still very much driven by information retrieval and named-entity recognition technologies and dependent on large ontologies.

As conceived, question answering was a knowledge- and labor-intensive proposition. In particular, it required the construction of elaborate ontologies that encode mappings from question types to semantic classes of expected answers. Furthermore, elaborate answer type ontologies must be supported by specific named-entity detectors for each semantic class (often created by hand). A tight coupling between the two components meant that one would not be very useful without the other. For example, knowing that the expected answer type is "French artist" would not be useful without the ability to identify instances

of French artists in free text. To give a concrete example, FALCON [Harabagiu et al. 2000a], Southern Methodist University's entry in the TREC 2000 question answering track, employed 27 named-entity categories. The system had 15 top-level nodes in its manually constructed answer type hierarchy and complex many-to-many mappings between named-entities and answer types. Similarly, Webclopedia [Hovy et al. 2000], ISI's TREC entry in the same year, used a complex typology with 94 total nodes, including 47 leaf nodes. This was created after manual analysis of 17,384 questions mined from online sources. Both were among the top ranking systems in the TREC 2000 question answering track, which validated the effectiveness of this approach. However, the knowledge-intensive nature of this solution posed a high barrier of entry for new researchers. Furthermore, the custom nature of these knowledge structures and tight coupling of system components discouraged sharing of resources and processing modules.

As a result of these circumstances, other viable approaches to factoid question answering became a focus of research at the beginning of this decade. Were there any shortcuts to circumvent the enormous costs associated with massive ontology-building? As it turned out, the property of data redundancy and the enormous amounts of text available on the Web provided a partial solution. Although redundancy-based systems do not perform as well as traditional knowledge-driven systems (of the type described above), their performance is nevertheless respectable. The contrast between traditional and redundancy-based approaches is a theme that will be continually explored throughout this work. See in particular Section 9.1 for a broader view of different approaches to factoid question answering.

Although Web-based question answering systems have existed since 1993,[1] the explosion of Web content sparked renewed interest at the beginning of this decade in techniques for mining answers from the Web. At first, the Web was simply used as another corpus; the MULDER system [Kwok et al. 2001], for example, relied on natural language parsing of both questions and potential answers, much like many other systems at the time. Unfortunately, MULDER was never formally evaluated at TREC, making direct comparison of results difficult.

As the Web steadily grew in size and importance, researchers starting with Kwok et al. [2001] realized that extracting answers from the Web necessitated a different approach. The enormous quantities of data available presented new challenges and opportunities for question answering: although the Web is orders of magnitude larger than any existing manually collected research corpus, the quality of each individual document is much lower. Furthermore, developing the software and hardware infrastructure for crawling and indexing the Web represents a serious undertaking.

One approach to Web-based QA, developed by Clarke et al. [2001a, 2001b] at the University of Waterloo, involved using the Web as a secondary source to validate answers extracted from a primary, more authoritative, corpus. Another

[1]MIT's START [Katz 1997] system holds the distinction of being the first question answering system *on* the Web; it came online in December 1993.

approach, pioneered by Brill et al. [2001, 2002] at Microsoft Research (the team included the present author), employed the Web as the sole source of answers [cf. Dumais et al. 2002]. They developed a set of techniques that took advantage of "snippets" from existing search engines, employing neither ontological resources nor any named-entity recognizers. The system developed by Microsoft Research, AskMSR, performed surprisingly well at the 2001 TREC question answering evaluation and generated energetic discussions among participants at the workshop. Their approach leveraged a characteristic of the Web dubbed data redundancy, and has come to be known as the redundancy-based approach. Since its introduction, this technique has gained widespread popularity and acceptance within the community. More recently, the AskMSR system has been augmented with Bayesian network models for predicting the likelihood that an answer is correct [Azari et al. 2004].

Aranea is a complete reimplementation, with additional refinements, of the original AskMSR system. Its modular architecture facilitates controlled experiments and the development of additional capabilities. In contrast, AskMSR was a monolithic, ad hoc agglomeration of two separate systems[2] whose output was then stitched together by yet another "combiner" system. Although the end-to-end performance of redundancy-based techniques is known through TREC evaluations, few studies have examined the performance contributions of individual components in detail. Aranea's architecture makes ablation studies easy to conduct, allowing one to isolate the impact of different modules and parameter settings. This work examines the various factors that affect factoid question answering and attempts to answer the question "What really matters?" The next section discusses the principle of data redundancy, which serves as the basis for the two main theses articulated in this article.

2.2 Data Redundancy

At the highest level, data redundancy allows systems to capitalize on statistical regularities to extract "easy" answers to factoid questions from the Web. This section elaborates in more detail.

The process of answering questions requires a system to make some sort of connection between the question and document passages containing answers. The task is relatively simple if these connections are manifested at the lexical level, that is, if there is a large degree of term overlap. However, this is often not the case, as the richness and expressivity of natural language allows humans to generate a variety of different textual forms that share the same semantics (thus containing answers), yet bear little superficial resemblance to each other and the question. To illustrate, consider the following questions paired with possible answers:

—*When did Alaska become a state?*

(1) Alaska became a state on January 3, 1959.
(2) Alaska was admitted to the Union on January 3, 1959.

[2]Affectionately known as "AskBrill" and "AskLin."

—*Who killed Abraham Lincoln?*

(1) John Wilkes Booth killed Abraham Lincoln.

(2) John Wilkes Booth altered history with a bullet. He will forever be known as the man who ended Abraham Lincoln's life.

—*When did Wilt Chamberlain score 100 points?*

(1) Wilt Chamberlain scored 100 points on March 2, 1962, against the New York Knicks.

(2) On December 8, 1961, Wilt Chamberlain scored 78 points in a triple overtime game. It was a new NBA record, but Warriors coach Frank McGuire didn't expect it to last long, saying, "He'll get 100 points someday." McGuire's prediction came true just a few months later in a game against the New York Knicks on March 2.

In all three cases, both passages do indeed contain answers. However, it seems obvious that a computer system could more easily extract the correct answer from the passages in (1) than the passages in (2). The task of answering factoid questions becomes much easier if the document collection contains answers stated using the same words and phrases found in the question.

Unfortunately from the point of view of text processing algorithms, answers can be expressed in a variety of different ways. Within a given text collection, it is possible that answers to the above questions are only stated in the passages marked (2). In other words, the only passages that contain answers may be those in which they are not obviously stated. In these cases, since the answer passages share few words in common with the questions, sophisticated natural language processing may be required to relate them, for example, recognizing syntactic alternations, collapsing paraphrases, resolving anaphora, making commonsense inferences, performing relative date calculations, etc.

The redundancy-based approach to factoid question answering leverages the massive amounts of data on the Web to avoid processing "tough answers." Because each item of information is likely to have been stated multiple times, in multiple ways, in multiple documents, there are many opportunities for extracting "easy answers." This property of the Web can serve as a surrogate for sophisticated language processing capabilities.

But how can a system automatically pinpoint the answer within a passage of text that contains terms from the question? Application of named-entity recognizers seems like an obvious solution, but this requires external knowledge about different answer types and their mappings to question types—essentially, an ontology of some sort, which is what we've been avoiding in the first place. However, data redundancy provides another potential solution: the correct answer is likely to have a high statistical correlation with terms from the question, which we can observe across many passages. Aranea implements this idea in the form of n-gram generation and subsequent processing of these n-grams.

Beyond statistical associations between query terms and answers, researchers hypothesize that the enormous size of the Web allows the use of simple pattern matching techniques to directly extract answers. With enough data, there is a good chance that an answer will appear as a simple reformulation of the question. In such cases, the answer could be extracted by directly

searching for an anticipated answer form, for example, by searching for the strings "Alaska became a state," "killed Abraham Lincoln," "Wilt Chamberlain scored 100 points," and extracting words that occur either to the immediate right or left. Naturally, this simple technique depends critically on the collection having an answer formulated in a specific way. The larger the collection, the more likely it is to have answers stated in more ways. Thus we can see that the Web, which is the largest known collection of text, has the greatest potential to contain answers in exactly the anticipated form. This simple pattern-based approach has been used by many other factoid question answering systems [Ravichandran and Hovy 2002; Fleischman et al. 2003; Hildebrandt et al. 2004], as well as in information extraction applications [Agichtein and Gravano 2000; Cafarella et al. 2005].

The above discussion presents two distinct ways that data redundancy can be leveraged for factoid question answering: simple pattern matching and statistical correlation between answer and question terms. What is the relative effectiveness of each strategy? This and other related questions will be empirically addressed in later sections.

Data redundancy also serves as a form of quality control. Because the quality of Web documents is lower than that of typical research corpora (i.e., newspaper articles), any single instance of a candidate answer is inherently untrustworthy. However, because a fact is usually found in multiple documents, a question answering system could utilize the distribution of answers across multiple sources to assess its reliability (for example, via a simple voting procedure). Note that such techniques often equate the *most popular* answer with the *correct* answer, which of course is not always the case (and often leads to humorous results).

The general effects of data redundancy have been noticed by many researchers. For example, Light et al. [2001] observed a correlation between the number of times an answer appeared in the TREC corpus and the average performance of TREC systems on that particular question. That is, systems tended to perform better on questions that had multiple answer instances within the corpus. Similarly, Clarke et al. [2001a] noticed an upward trend in question answering accuracy as a system was given more and more text from which to extract answers (holding everything else constant). Although these results seem intuitive, this work aims at a more refined quantitative understanding of this property.

The redundancy-based approach to factoid question answering was developed as an alternative to the traditional ontology-driven knowledge-based techniques implemented by other contemporary systems. At first glance, this approach appeared to embody the philosophy of "data is all that matters." Given enough data, a system could simply count instances and derive answers from these observations. This approach to tackling problems in language processing was demonstrated in a paper by Banko and Brill [2001]. They showed that performance in a supervised machine learning task increased with more training data—the relationship was approximately log–linear in the size of training samples. They also observed that the amount of training data had a more significant impact on performance than the specific choice of machine learning algorithm (and the specific task, for that matter). Given enough data, simple

counting performed about as well as any other learning method. Furthermore, noisy training data did not appear to have a detrimental impact on performance. In closing, Banko and Brill [2001] suggested that the availability of data is the single most important driving force in language processing applications (and by implication, more effort should be devoted to data collection).

Although it appears that redundancy-based methods exemplify this data-driven philosophy, this work attempts to develop the thesis that a lot more is going on: there are many other factors affecting factoid question answering performance that have nothing to do with data redundancy. In fact, there is a significant amount of "knowledge" that is encoded in redundancy-based techniques, in the form of heuristics—knowledge that is very different from formal question and answer type ontologies. Nevertheless, these components have a large impact on a system's ability to accurately mine answers from the Web.

Data redundancy also motivates the other major thesis explored in this work. To take advantage of this property, one must tackle the significant engineering challenges associated with manipulating enormous quantities of text. Fortunately, the nontrivial task of providing access to the terabytes of data on the Web has already been successfully solved by modern Web search engines. These existing services can be employed to provide a more manageable working collection of text, much in the same way that many traditional factoid QA systems use off-the-shelf document retrieval engines. However, there are important differences: whereas the exact underlying scoring algorithm of an IR engine is usually known, the scoring algorithm of a Web search engine is a closely guarded secret, and even worse, changes unpredictably over time. Thus, researchers are forced to either tackle the enormous engineering undertaking of crawling and indexing the Web or face the unpleasant reality of building systems on what essentially amount to black boxes. The first approach is beyond the resource capabilities of most research groups, while the second is unpalatable because it goes against well-established engineering sensibilities. The ability to decompose complex systems into simpler components is usually based on the assumption that one can control the specifications of individual components, or failing that, at least have a clear understanding of their behavior at the input/output interfaces.

Aranea relies on Web search engines it has little control over and whose internal workings it has little knowledge about. Yet the system is able to achieve good accuracy in answering factoid questions. This work develops the thesis that some characteristics of the Web are relatively stable, and the qualitative behavior of redundancy-based techniques appear to remain invariant with respect to many factors. This is demonstrated by a number of Aranea experiments that probe the behavior of Web search engines. In other words, the phenomenon of data redundancy appears to exist independently of specific implementation idiosyncrasies. As a result, it is possible to build reliable factoid systems that use external "black box" components.

3. PERFORMANCE OF REDUNDANCY-BASED APPROACHES

How well do redundancy-based techniques work in practice? The NIST-organized TREC question answering tracks provide a benchmark. These

evaluations, which started in 1999, have become the de facto standard for objectively quantifying question answering accuracy. The annual forums provide the infrastructure and support necessary to conduct large-scale evaluations on shared collections using common, blind test sets, thereby providing meaningful comparisons between different retrieval techniques. The TREC model has been duplicated and elaborated on by CLEF [Magnini et al. 2004] in Europe and NTCIR in Asia [Fukumoto et al. 2002], both of which place additional emphasis on cross-lingual aspects of question answering.

Redundancy-based techniques were first implemented by Microsoft researchers (including the author of the present article) in the AskMSR system [Brill et al. 2001, 2002], which participated in the TREC 2001 QA evaluation. Presentation of TREC results, however, must be prefaced by an explanation of the evaluation methodology.

The implementation of the QA task by NIST calls for system responses composed of [*docid*, *answer string*] pairs. That is, answers must be "supported" by a document selected from the target corpus of newspaper and newswire articles.[3] Support meant that a human reading the document must be able to "figure out" the answer; if a document coincidentally contained the correct answer string, the response would be marked "unsupported." As a concrete example, consider the question "What Spanish explorer discovered the Mississippi River?" A response of "Hernando de Soto" paired with a document that contained the fragment "the 16th-century Spanish explorer Hernando de Soto, who discovered the Mississippi River . . . " would be judged as correct. However, the same answer string, paired with a document that contained the sentence "In 1542, Spanish explorer Hernando de Soto died while searching for gold along the Mississippi River" would be judged as "unsupported." Since AskMSR extracted answers directly from the Web, participation in TREC required an extra step to "project" Web-derived answers back onto the target corpus (to find a valid supporting document). One could reasonably argue that, for Web-based systems, answer projection is a merely an artifact of the TREC QA setup.

Based on the lenient evaluation measure, where unsupported answers are counted as correct, the AskMSR system achieved a mean reciprocal rank (MRR) of 0.43; it was unable to find a correct answer in the top five responses for 40% of the questions. In the evaluation setup, each question is given a score equal to the reciprocal of the rank at which the first correct response was found, or 0 if none were found. Thus, a correct response would receive 1 at rank one, 1/2 at rank two, 1/3 at rank three, etc. The mean reciprocal rank is simply the average across individual questions' reciprocal ranks. Overall, AskMSR ranked sixth out of thirty-six participants. In contrast, the system fared worse when considering the strict evaluation measure, where unsupported answers were considered wrong. It ranked ninth out of thirty-six teams, achieved an MRR of 0.35, and was unable to find a correct answer in the top five responses for 49% of the questions.

As can be seen, answer projection has a significant impact on performance. For approximately 20% of the questions, the AskMSR system was unable to find

[3]See Voorhees [2001, 2002] for detailed descriptions of the corpora.

Table I. Performance of Aranea at TREC 2002 and TREC 2003

	Strict Scoring		Lenient Scoring	
Year	Accuracy	Rank	Accuracy	Rank
2002	.304	8/34	.456	5/34
2003	.295	6/25	.383	5/25

an appropriate supporting document for an otherwise correct answer. It was clear that the simplistic answer projection algorithm employed by the system, which used both the question and the top Web-derived answer as a query to an off-the-shelf document retrieval engine, was insufficient to find good supporting documents.

The performance of AskMSR is noteworthy considering that the system took less than 2 months to develop. The original motivation of the redundancy-based approach was to shortcut the massive knowledge engineering effort required to build high-accuracy factoid question answering systems. With respect to this goal, the system succeeded. Although it did not beat the most sophisticated ontology-driven systems (many of which had been under development for several years), the performance of relatively simple redundancy-based techniques was surprising.

The track record of redundancy-based techniques continued with Aranea in TREC 2002 and TREC 2003 (see summary of performance in Table I). Starting in 2004, NIST altered the evaluation methodology to reflect the evolving interests of the research community: instead of individual factoid questions that could be understood in isolation, the basic evaluation unit became a question series, which included factoid, list, and "other" questions arranged around a central topic, or *target* [Voorhees 2004]. Since the factoid questions were no longer self-contained, evaluation of factoid QA accuracy in isolation became more difficult.

In the 2002 and 2003 TREC QA evaluations, [*docid*, *answer string*] response pairs were assigned one of four judgments: correct, incorrect, unsupported, or inexact. The notion of answer exactness requires some explanation: an answer string was considered inexact if it contained extraneous words that were not necessary to answer the question. Inclusion of any additional material in the answer string (e.g., justification) would render it inexact. In our example about the discoverer of the Mississippi River, "Hernando de Soto" would be exact, whereas "the Spaniard Hernando de Soto" would not be. Under this scoring methodology, Aranea correctly answered 30.4% of all questions in TREC 2002 and 29.5% of all questions in TREC 2003 (see Table I). Starting in 2002, systems could only return a single response per question, so the relevant metric is answer accuracy. Overall, Aranea ranked eighth in 2002 and sixth in 2003, out of 34 and 25 participants, respectively. Recognizing that answer support can be viewed as an artifact of the TREC setup, and that inexact answers are still useful, Table I also reports a lenient scoring condition, where inexact and unsupported answers are counted as "correct." Under this scoring method, Aranea's answer accuracy jumps to 45.6% in TREC 2002 and 38.3% in TREC 2003, which ranked it fifth in both years.

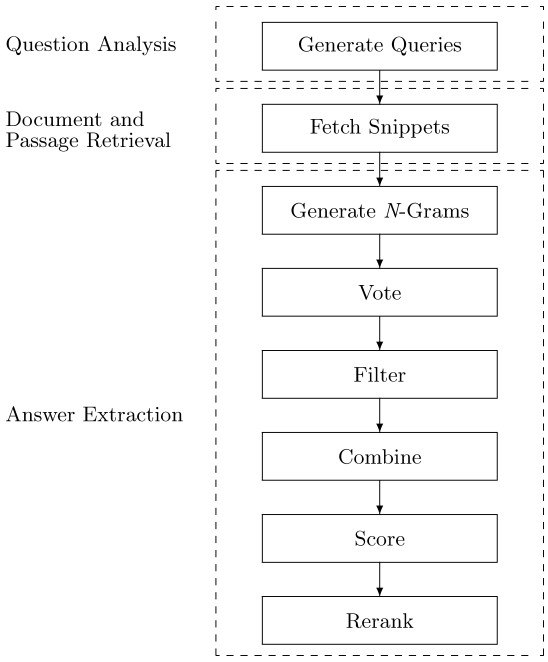

Fig. 2. Overview of Aranea's processing pipeline.

Since the TREC QA tracks only evaluate end-to-end performance, it is difficult to determine what factors contributed to the success of redundancy-based techniques (both in the AskMSR system and in Aranea). The rest of this article explores the many factors that affect factoid question answering performance through a series of contrastive experiments and ablation studies.

4. ARANEA ARCHITECTURE

At the highest level, Aranea operates by mining summaries of Web pages returned by commercial search engines. These text snippets are manipulated by a series of modules in a processing pipeline. The final output is a ranked list of answers to the original factoid question.

Aranea's system architecture is shown in Figure 2 [Lin and Katz 2003]. This arrangement of modules shares many common features with the traditional question answering architecture illustrated in Figure 1:

—Both architectures contain what can be generalized as components for question analysis. In the redundancy-based approach, this component does not rely on sophisticated language processing capabilities or ontological resources.

—Both architectures contain components for retrieval. Whereas existing off-the-shelf IR engines are used by traditional QA systems, the equivalent task is handled by Web search engines in Aranea.

—Both architectures contain what can be generalized as components for answer extraction. However, redundancy-based systems do not rely on named-entity detection or other language processing capabilities.

These architectural similarities at the functional level stem from the same logical decomposition of the question answering task: analyze the question, fetch relevant text, and then pinpoint the exact answer. In the following sections, each Aranea module will be described in detail. To better illustrate the question answering process, we will consider the following two factoid questions:

—What year did Alaska become a state?

—Who was the first person to run the mile in less than four minutes?

4.1 Generating Queries and Fetching Snippets

The first step in answering factoid questions is to fetch text snippets that potentially contain the answer. This is accomplished by formulating queries to existing Web search engines (Google and Teoma, in our case) and extracting the search results—this is accomplished through custom wrappers for each site. Instead of processing the actual HTML contents of retrieved hits, Aranea operates entirely on the keyword-in-context summaries generated by each search engine. In effect, these short text segments represent relevant passages extracted from the retrieved hits.

In addition to the baseline query, which is simply the verbatim natural language question, Aranea generates two other types of queries: exact and inexact reformulations. Queries of all three types are concurrently generated, but exact reformulations are weighted five times more than the other two types (by default).

An exact query anticipates the specific location of an answer and attempts to directly extract it using surface patterns. For example, Aranea would determine that the answer to the question "When was the telephone invented?" is likely to appear to the right of the exact phrase "the telephone was invented"; this yields the exact reformulation "the telephone was invented **?x**", where **?x** indicates the location of the anticipated answer. Similarly, "the Valley of the Kings is located in **?x**" is an exact reformulation of the question "Where is the Valley of the Kings?" Exact reformulations translate into phrase queries in Google and Teoma, followed by pattern matching on the snippet texts to extract the unbound variable. Since the system has no method for predicting the length of an answer (without more sophisticated linguistic analysis), the variable matches five tokens (up to 50 characters) by default.

Exact reformulations are generated by approximately a dozen pattern matching rules based on question terms and their part-of-speech tags. Pattern matches at the morpho-lexical level trigger the creation of reformulated exact queries. As an example, the first query was generated by the rule "*wh-word* did A *verb* B → . . . A *verb*+ed B **?x**". An internal lexicon ensures that the generated verb receives the proper morphology (handling irregular forms, for example). Aranea's reformulation rules range from the general, as in the above example,

What year did Alaska become a state?	
[baseline]	What year did Alaska become a state
[inexact]	Alaska became a state
[exact]	Alaska became a state ?x

Who was the first person to run the mile in less than four minutes?	
[baseline]	Who was the first person to run the mile in less than four minutes?
[inexact]	the first person to run the mile in less than four minutes
[exact]	the first person to run the mile in less than four minutes was ?x
[exact]	?x was the first person to run the mile in less than four minutes

Fig. 3. Example queries generated by Aranea.

to those specific to particular question types, for example, "Where is $A \rightarrow A$ is located in **?x**". Some reformulations, like the first example, manipulate only the sequence of query tokens and their morphology, while others, like the second example, insert additional words not present in the original questions.

The triggering of each reformulation rule produces an exact reformulation and a corresponding inexact reformulation, where the query terms are treated as a bag of words (i.e., no matching of the unbound variable is performed). These queries gain the benefit of expanded (and reordered) query terms, but retain broader coverage because an exact match of the reformulated phrase is not required.

As two complete examples, the queries generated for the questions "What year did Alaska become a state?" and "Who was the first person to run the mile in less than four minutes?" are shown in Figure 3. Aranea's reformulation rules use part-of-speech tags that have been assigned by Brill's tagger [Brill 1995]:

—What/WP year/NN did/VBD Alaska/NNP become/VB a/DT state/NN
—Who/WP was/VBD the/DT first/JJ person/NN to/TO run/VB the/DT mile/NN in/IN less/JJR than/IN four/CD minutes/NNS

By default, Aranea attempts to fetch 100 snippets per query. Typically, however, exact reformulations yield substantially fewer results. Also by default, the **?x** in exact reformulations matches five tokens (up to 50 characters).

A few samples of text snippets containing answers to the question "What year did Alaska become a state?" are shown in Figure 4. Text snippets gathered from the result of different queries are assigned a score based on the importance of that particular query. By default, exact queries are given five times more weight than inexact and baseline queries. Because multiple queries may fetch the same snippets, duplicates are removed by using the URL as a unique key. To facilitate repeatable experiments, Aranea implements a caching mechanism for Web pages.

As discussed in Section 2.2, there are two general approaches to leveraging data redundancy: exploiting the statistical associations between question and answer terms, and directly extracting answers using reformulation patterns. The first is represented by the baseline queries and inexact reformulations,

Baseline Query

—...become a state until January 3, 1959. Today, the tourism industry in Alaska alone generates over $1.4 billion in revenues each year and will...

—When did Alaska become a state? January 3, 1959 What number state to enter the union? 49th ... some permafrost, ground that is frozen all year.

—The year was 1959. Alaska would become the 49th state, and Hawaii would become the 50th. "Ben-Hur" and "Some Like ...

Inexact Reformulation

—The strategic importance of Alaska was finally recognized in World War II. Alaska became a state on January 3, 1959.

—They also say that Alaska, which became a state in 1959, needs more money to catch up with the rest of the country in developing its...

—Alaska became the 49th state in January 1959. But the United States contends it laid claim to the submerged land when Glacier Bay National...

Exact Reformulation

—Since Alaska became [a state in 1959, it] has gotten 6.6 times more federal road money than it has contributed in gas taxes.

—The capital city of Alaska is Juneau. Alaska became a State [on May 16, 1959 (49th).] Alaska's motto is "North To The Future".

—But when Alaska became a state [in 1959, Congress gave the] new state rights to about 104 million acres. Then, in 1971, Congress settled Alaska...

Fig. 4. Sample snippets retrieved by different query types for the question "What year did Alaska become a state?" Brackets denote the extent of the text that matches the query pattern in the exact reformulation.

while the second is represented by exact reformulations. The relative effectiveness of these two approaches is an empirical question we explore later.

Aranea's query formulation module translates baseline, inexact, and exact queries into syntactically valid Google and Teoma queries, expressed in terms of the appropriate query modifiers, for example, quotes in the case of exact queries. Nevertheless, the system has limited control over the results. For example, snippets returned in response to a quoted Google query are not guaranteed to have the query phrase verbatim. Query terms may not be present in a snippet despite the presence of a plus operator in front of the term (the term may be present in the document, but not within the snippet window). Both Google and Teoma will display up to 100 hits per page; in order to fetch more results, Aranea simulates clicking the "next page" button by fetching the URL associated with that link. This process is repeated until the desired number of snippets is fetched or until the search engine runs out of hits. As will be discussed in Section 6.2, both Google and Teoma place a limit on the number of results that can be retrieved in this manner, even though they report many more matching Web pages. In short, these external components can be characterized as "black boxes" whose behavior at the input/output interfaces are unknown, and to a certain extent, unpredictable. Exploring the implications of this reality is one of the major goals of this work.

4.2 Generating n-Grams

Answers to TREC questions are almost always noun phrases (and, very rarely, verb phrases). However, Aranea does not rely on natural language processing technology to identify valid noun phrases or any other linguistically cohesive answer units. Because the system must often manipulate snippets of text that may not be well-formed natural language to begin with, we have adopted a robust strategy for enumerating answer candidates based on n-gram generation. Such an approach has empirically proven to be effective.

This Aranea module exhaustively generates all unigrams, bigrams, trigrams, and tetragrams from the text snippets retrieved by the previous module. Aranea does not generate n-grams across segment boundaries within snippets (denoted by " ... "), since they represent noncontiguous portions of a Web page. These n-grams, which are given scores equal to the weight of the query that retrieved them, serve as initial candidate answers.

4.3 Voting

The voting module collates the n-grams generated by the previous module. The new score of each answer candidate is equal to the sum of the scores of all occurrences of that particular n-gram. Therefore, higher scores are assigned to text fragments that occur more frequently in the retrieved snippets. These n-grams are more likely to denote correct answers.

The top-scoring candidate answers after voting are usually stopwords and query terms themselves. As a concrete example, the highest-scoring candidates after the voting process for the question "What year did Alaska become a state?" are "the," "Alaska," "in," "state," "of," "a," and "1959" (the correct answer), in that order. For the question "Who was the first person to run the mile in less than four minutes?" terms from the question and stopwords similarly occupy the highest-ranking positions. Fragments of the correct answer, "Bannister," "Roger," and "Roger Bannister" are ranked 26th, 35th, and 38th, respectively. Subsequent processing will eliminate obviously incorrect answers and promote correct answers to the top of the candidate list.

4.4 Filtering

After voting, answer candidates are passed through a series of coarse-grained filters meant to discard n-grams that are obviously wrong. Aranea applies three types of filters: type-neutral, type-specific, and closed-class.

Since this module outright discards certain answer candidates, the heuristics almost never throw away possibly correct answers, but let many wrong answers through. Incorrect answers that remain can be sorted out by modules downstream in the processing pipeline, but the system will not be able to recover from good answers incorrectly thrown away.

The first type of filter implements heuristics that are neutral with respect to the question type:

—Candidates that begin or end with stopwords are discarded.

—Candidates that contain words found in the original natural language question are discarded. The only exception is question focus words, for example,

a question beginning with "How many *meters* ... " can be answered by an expression containing the word "meters." Although one could imagine rare instances where answers contain terms found in the question, for example, "Who was New York named after?" this has not been observed in the TREC test sets.

The second type of filter implements a series of rules that are question-type specific. Certain surface cues restrict the range of possible answers, and are captured in heuristics meant to decrease the number of spurious candidates. For example, the answer to "how far," "how fast," "how tall," etc., questions must contain a numeric component (either numeric digits or written numbers); thus answer candidates that do not fit this criterion can be safely discarded. "Who" and "where" questions generally require proper noun answers. As a result, answer candidates whose first and last tokens do not start with a capital letter are discarded. Aranea does not require all tokens within a candidate to have an initial capital letter, since many correct proper noun answers contain articles, prepositions, etc. Manual analysis of the TREC test sets confirms the intuition encoded by this heuristic: indeed, almost all "who" and "where" questions are answered by proper nouns. The only exceptions are examples such as "Where is your corpus callosum?" These are properly handled (by pure happenstance) because common nouns exist with their initial letters capitalized on the Web, for example, "Brain."

Finally, the third type of filter applies to questions whose possible answers can be exhaustively enumerated. For example, a question such as "What language do most people speak in Brazil?" must be answered with a language; thus, we can safely throw out any answer candidate that isn't a known language. For a variety of question types, for example, "What country ... ," "What state ... ," "What sport ... ," "What nationality ... ," it is possible to enumerate all acceptable answer candidates. In these cases, fixed lists can be used as high-precision filters to discard candidates not known to be correct. Aranea implements filters for 17 such answer types. These closed-class filters contain enumerated entities that can be readily mined from the Web.

This Aranea module encodes a significant amount of heuristic knowledge about "what makes a good answer." In fact, it serves much the same purpose as elaborate answer type ontologies in traditional factoid systems. What is the relative impact of these heuristics? In other words, to what extent is data redundancy the sole driver of question answering performance, or does knowledge play an important role also?

To continue with concrete examples, the filtering module applies the year filter to retain the correct answer to the question "What year did Alaska become a state?" which is "1959." The year filter discards all candidates that are not four digit numbers (but retains numbers explicitly marked with "AD," "BC," etc.). Since "1959" was the highest-scoring candidate retained by the filter, it was promoted to the top answer position.

For the question "Who was the first person to run the mile in less than four minutes?" the filtering module discards candidates that contain stopwords or terms from the question, as well as candidates with leading or trailing lowercase

After filtering

Candidate	Score
Bannister	137
Roger	114
Roger Bannister	103
English	26
...	...

After combining

Candidate	Score
Roger Bannister	354
Sir Roger Gilbert Bannister	286
Sir Roger Bannister	280
Bannister Sir Roger	278
...	...

After scoring

Candidate	Score
Roger Bannister	2377
Englishman Roger Bannister	1853
Sir Roger Gilbert Bannister	1775
Sir Roger Bannister	1768
...	...

Fig. 5. Answer candidates to the question "Who was the first person to run the mile in less than four minutes?" after the filtering, combining, and scoring modules.

terms (since "who" questions require proper noun answers). The top-scoring answer candidates after this filtering process, as well as their scores, are shown in Figure 5.

4.5 Combining

In the combining module, unigrams are used as evidence to boost the score of longer answer candidates. This technique is employed to counteract the tendency of the n-gram generation process to favor shorter, but more frequently occurring candidates. The score of a candidate answer S_c is incremented by the sum of the scores of its component unigrams:

$$S_c = S_c + \sum_{t \in c} S_t.$$

For example, "Roger Bannister" would receive a score boost from the unigrams "Roger" and "Bannister." In essence, this scoring heuristic serves as a chunker that identifies coherent multiword phrases present in the textual snippets based on the relative frequency of different candidates and their constituent unigrams. The combining module guards against the promotion of nonconstituent candidates with extraneous words, such as "Roger Bannister ran" or "was Roger Bannister" in a variety of ways:

—The filtering module (Section 4.4) discards candidates that are known to be invalid multiword expressions. For example, n-grams with leading or trailing stopwords are removed. For questions that require proper noun answers,

n-grams with leading or trailing lowercased terms are discarded. As a result, the combining module will not need to deal with these candidates.

—The score of the full candidates themselves are taken into consideration by the combining heuristic. For example, consider the n-gram "Roger Bannister Completed": it would receive score contributions from the unigrams "Roger," "Bannister," and "Completed." However, since the n-gram "Roger Bannister Completed" occurs far less frequently than "Roger Bannister," its final score would be lower, even though it received additional score contributions from the unigram "Completed."

—As we will discuss in Section 4.6, candidate answers with extraneous terms will tend to have lower scores based on Aranea's *idf*-based scoring method.

To continue with our example, the top-scoring candidates for the question "Who was the first person to run the mile in less than four minutes?" are shown in Figure 5.

4.6 Scoring

Up to this point in the processing pipeline, there is an implicit assumption that every term represents an equally likely observation within the text snippets mined from the Web. However, it is a fairly obvious fact that terms do not have equal prior distributions, and that not all terms are of equal importance. All things being equal, answer candidates comprised of less common words should be favored over answer candidates comprised of more common words.

In Aranea, this intuition is captured by inverse document frequency (*idf*), a long-standing metric used in information retrieval [Robertson 2004]. Aranea computes the average of the *idf* of the unigrams that comprise a particular candidate answer. The score of a candidate answer is then multiplied by this value. However, since it is difficult to determine the exact distribution of terms on the Web, statistics from the AQUAINT collection[4] are used as a surrogate. This corpus, which consists of approximately one million articles from *The New York Times*, the Associated Press, and the Xinhua News Agency totaling around 3 GB, has been used as the official corpus of the TREC question answering tracks since 2002.

In more detail, each answer candidate S_c is scored in the following manner:

$$S_c = S_c \times \frac{1}{|c|} \sum_{w \in c} \log\left(\frac{N}{w_{cnt}}\right).$$

The variable c denotes the set of terms in the candidate answer; N is the total number of documents in our collection; w_{cnt} is the number of documents in the collection that contain the word w.

The scoring module has the additional impact of demoting candidate answers that contain extraneous words. For example, "Roger Bannister Completed" has a lower unigram *idf* average than "Roger Bannister." In general, named-entities

[4]The AQUAINT Corpus of English News Text (LDC2002T31), distributed by the Linguistic Data Consortium, available online at `http://www.ldc.upenn.edu/`.

will tend to have higher scores relative to nonconstituent answer candidates that contain those named-entities.

Figure 5 shows the top-ranking answer candidates to the question "Who was the first person to run the mile in less than four minutes?" after the scoring module. "Roger Bannister," the correct answer, remains at the top, but the ordering of other answer candidates has been affected.

4.7 Reranking

At the end of the processing pipeline, Aranea employs another set of heuristics designed to detect likely correct answers for a few specific question types; these recognized answer forms are pulled to the top of the ranked list of answers. This reranking process is fundamentally different from the filters discussed in Section 4.4. The reranking heuristics select likely correct answers instead of throwing away incorrect ones. For example, certain answer forms of "when" and "where" questions are easy to recognize; if one such form is found in the top five answer candidates, Aranea brings it into the top-ranking position. Recognized answer forms of "when" questions include standard combinations of month, day, and year; recognized answer forms of "where" questions include "city, state" combinations. Since these heuristics do not anticipate all possible answer forms, applying them at the filtering stage would likely discard many correct answers.

Since selecting a wrong answer is often worse than producing no answer at all, Aranea only returns answers that are supported by at least two unique text snippets mined from the Web. The system iterates through candidate answers and counts the number of snippets that contain it; answers for which the count is less than two are discarded. If this process eliminates all possible answers, Aranea returns "don't know."

This component, as with the filtering module, encodes a nontrivial amount of knowledge about the relationship between different questions and potential answers. Determining the impact of these heuristics is one key to understanding the relationship between knowledge and data redundancy.

5. EXPLORATION OF REDUNDANCY-BASED TECHNIQUES

The redundancy-based approach to factoid question answering, while conceptually simple, breaks down into many individual techniques and components. The main purpose of this work is to explore the many factors that affect performance. Through ablation studies and one-off runs, we hope to untangle the contribution of the various modules, external components, and parameter settings. With Aranea, such experiments are easy to conduct due to its modular architecture.

Specifically, this exploration of redundancy-based methods is couched within the context of two overarching questions, summarized from Sections 1 and 2.2:

—Although redundancy-based techniques represent a data-driven approach, there is nevertheless a significant amount of knowledge that is encoded in the form of heuristics. What is the impact of this knowledge?

—The most practical workaround to the challenge of crawling and indexing the entire Web is to leverage output from existing Web search engines. However, this creates the uncomfortable situation where the question answering system is dependent on external components with unknown specifications. How can a system cope with such realities?

Following a more logical bottom-up order, the second issue is tackled first. We conducted a series of experiments to probe the behavior of Web search engines and uncovered a number of stable characteristics. These appear to be generalizations about data redundancy, as opposed to idiosyncratic properties of system implementations or parameter settings. Capitalizing on these observations, it is possible to build functional factoid QA systems that rely on Web search engines. Section 6 describes experiments that attempt to address the following questions:

—What is the effect of using different search engines?
—How does answer accuracy vary with the number of snippets mined?
—How does answer accuracy change over time as the content of the Web evolves?

The role of "knowledge" in redundancy-based factoid question answering is explored in Section 7. Although the knowledge encoded in various Aranea modules is very different from the formal ontological structures employed by traditional systems, it nevertheless captures certain regularities exhibited by factoid questions and their answers. Experiments reveal that this knowledge has a significant impact on question answering performance. Specifically, the following questions are examined:

—How effective are different query types?
—How effective are surface patterns for directly extracting answers?
—How effective are the filtering and reranking heuristics?

Finally, there are a number of Aranea components and settings that do not directly contribute to either of the two theses mentioned above, but understanding their impact will provide a more complete picture of redundancy-based methods. The following questions are discussed in Section 8:

—How does answer accuracy vary with the size of the n-grams mined?
—How effective is the combining heuristic?
—How effective is *idf*-based scoring of candidate answers?

Naturally, it would be impractical to answer all these research questions if there were no methods for automatically evaluating the output of factoid QA systems. Fortunately, there exist resources for automatic system evaluation. Each year after TREC, Ken Litkowski of CL Research manually creates a set of regular expressions that captures the correct answer forms, based on human judgments of actual system responses. In addition, NIST provides a scoring script that matches answer patterns against system output. Although these resources do not serve as a true test collection [Voorhees and Tice 2000;

Table II. Summary of the Test Sets Used in Our Experiments.

Test Set	Original Size	Final Size
TREC-9 (2000)	500	440
TREC-10 (2001)	500	321
TREC-11 (2002)	500	454
TREC-12 (2003)	413	383

Lin 2005; Lin and Katz 2006], they are nevertheless useful for automatic evaluation.

All experiments described in this article employ the standard TREC answer patterns.[5] Note that these patterns do not check for supporting documents, so the scoring criteria is rather liberal. At the same time, however, these patterns are overly restrictive because they do not accept all possible correct answers. For example, the patterns do not capture numeric answers whose values change over time, such as, "What is the population of Mexico?" since answer patterns were crafted with respect to a static corpus. Some questions, for example, "When was the first flush toilet invented?" are open to interpretation, and hence Web data might support a different set of answers than the official TREC corpus. Answer patterns often do not adequately capture these divergences. Finally, regular expression answer patterns do not penalize answers with extraneous words, such as "Thomas Edison invented"—under official TREC scoring guidelines [Voorhees 2002], such responses would be considered "inexact." Due to these numerous reasons, the absolute performance of each system variant under different experimental conditions should not be taken in isolation; the figures are only meant to be interpreted with respect to a similarly-evaluated contrastive condition.

To tackle the research questions set forth, we employed test data from the TREC QA tracks, 2000 through 2003. All experiments were conducted in September 2005. Preparation of the test sets included additional filtering of the original questions. In some years, a few questions were dropped from the official evaluation due to spelling/grammatical errors, ambiguities, etc. These were discarded due to lack of corresponding answer patterns. Questions for which there were no known answers in the document collection were removed for the same reason. In addition, questions that asked for a definition (e.g., "What is bipolar disorder?"; "What is cholesterol?") were removed. This turned out to be a substantial number for the TREC-10 test set. After much discussion within the community, it was agreed that definition questions were substantially different from factoid questions and should be evaluated according to a different methodology [Voorhees 2003; Lin and Demner-Fushman 2005]. These questions were no longer included with factoids starting in 2002. A summary of the test sets used in our experiments is shown in Table II.

For each of the runs, the top five answers were automatically scored using the standard TREC answer patterns and scoring script. Three separate measures of answer accuracy were collected: mean reciprocal rank (MRR), the fraction of

[5]These can be downloaded at `http://trec.nist.gov`.

questions with correct answers at rank one (C@1), and the fraction of questions for which a correct answer was found in the top five responses (C@5).

All experiments described in this article compare a default condition[6] with one or more contrastive conditions. Unless otherwise specified, the default configuration of Aranea is the full system fielded in the official TREC evaluations, minus the component that utilizes semistructured data resources (see Section 5.1). Relative increases or decreases in each of the metrics are noted, as well as the statistical significance of the differences. For mean reciprocal rank, the Wilcoxon signed-rank test is used because it makes minimal assumptions about the underlying distribution of differences. With the metrics C@1 and C@5, the sign test is used due to the binary nature of the scores. For each evaluation metric, significance at the 1% level is indicated by either $^{\blacktriangledown}$ or $^{\blacktriangle}$, depending on the direction of change; significance at the 5% level is indicated by $^{\triangle}$ or $^{\triangledown}$, depending on the direction of change. Differences that are not statistically significant are marked with the symbol $^{\circ}$. Unless otherwise noted, this work reports results of system experiments performed on Web pages gathered in September 2005. Since Aranea implements a caching mechanism that allows the reuse of previously fetched Web pages, all runs can be replicated at any time.

It is important to note that generalizations about redundancy-based techniques can only be drawn insomuch as Aranea is a typical system that implements such an approach. It is impossible to exhaustively explore the parameter space in any single dimension (e.g., the number of different search engines, the different methods for scoring n-grams, etc.), much less all the factors that may potentially affect question answering performance. This work describes experiments that represent interesting points in the solution space, and we believe that our results do provide a deeper understanding of the redundancy-based approach. After presenting experimental results in Sections 6, 7, and 8, we will return to a higher-level discussion in Section 9.

5.1 The Role of Semistructured Database Techniques

The version of Aranea deployed in the official TREC evaluations employed semistructured database techniques as well as redundancy-based methods. Since the focus of this article is the latter approach, the effects the former must first be understood and quantified.

The use of semistructured database techniques for question answering is motivated by the observation that user questions follow a sort of Zipfian distribution—a small fraction of question *types* accounts for a significant portion of all question *instances*. Many questions ask for the same type of information, differing only in the specific object under consideration, for example, "What is the population of the United States?"; "What is the population of Mexico?"; "What is the population of Canada?" etc. Not only do such questions appear frequently, they can also be grouped together into coherent classes, for example, "What is the population of x?" where x is a variable that can stand for any country. A previous study has shown that 20 question types account for

[6]The term *baseline* is avoided because many experiments are ablation studies, so the default configuration actually performs the best.

Table III. Impact of Semistructured Database Techniques on Question Answering Accuracy

	TREC-9	TREC-10	TREC-11	TREC-12
MRR				
No lookup	.530	.582	.587	.448
Lookup	.543 (+2.5%)$^{\triangle}$	.622 (+6.9%)$^{\blacktriangle}$	.587 (+0.0%)$^{\circ}$	.449 (+0.2%)$^{\circ}$
C@1				
No lookup	.468	.517	.529	.392
Lookup	.482 (+2.9%)$^{\circ}$	.567 (+9.6%)$^{\blacktriangle}$	.526 (−0.4%)$^{\circ}$	.392 (+0.0%)$^{\circ}$
C@5				
No lookup	.627	.676	.678	.535
Lookup	.639 (+1.8%)$^{\circ}$	.698 (+3.2%)$^{\triangle}$	.678 (+0.0%)$^{\circ}$	.538 (+0.5%)$^{\circ}$

over 20% of questions from the TREC-9 test set and 40% of questions from the TREC-10 test set [Lin and Katz 2003]. Analyses of question logs from the START system [Katz 1997], which has answered millions of questions over the last decade, and query logs from commercial search engines [Lowe 2000] lead to the same conclusion.

Classes of commonly occurring questions suggest a view of question answering as database lookup, given an appropriate organization of Web data. It is indeed possible to organize Web resources into a vast semistructured database for question answering, as demonstrated by START [Katz 1997; Katz et al. 2002]—a much simplified version of the technology was implemented in Aranea and deployed in the TREC evaluations.

Results of Aranea runs with and without database lookup are shown in Table III. Since test sets from TREC-9 and TREC-10 were used to inform the knowledge engineering process necessary to structure the various Web resources, they must be viewed as development data. Since the system was frozen prior to TREC-11 (and no further development was pursued for TREC-12), questions from those years can be viewed as held-out test sets. As a result of this split, averaging answer accuracy across all four test sets is not meaningful. We see that the use of semistructured database techniques has negligible performance impact on the TREC-11 and TREC-12 test sets, beyond what can already be achieved with redundancy-based approaches alone. There seems to be an overlap in the types of questions covered by the two different approaches: the large classes of commonly-occurring questions that can be handled by database techniques appear to be the same types of questions that redundancy-based techniques can answer without difficulty. This conclusion, however, should not be extended to semistructured database techniques in general, given Aranea's limited coverage.

Our purpose is not to evaluate the effectiveness of semistructured database techniques, but to better understand its overall impact on the complete Aranea system deployed for TREC. Having accomplished this, the database components were disabled for all subsequent experiments.

6. THE IMPACT OF EXTERNAL COMPONENTS

The redundancy-based approach to factoid question answering is driven by access to large quantities of data provided by Web search engines. Although

Table IV. Impact of Different Search Engines on Question Answering Accuracy: All Query Types

	TREC-9	TREC-10	TREC-11	TREC-12
		MRR		
Both	.530	.582	.587	.448
Teoma	.502 (−5.3%)▼	.545 (−6.4%)▽	.550 (−6.3%)▼	.434 (−3.1%)◦
Google	.497 (−6.2%)▼	.532 (−8.6%)▼	.547 (−6.8%)▼	.400 (−10.7%)▽
		C@1		
Both	.468	.517	.529	.392
Teoma	.445 (−4.9%)◦	.495 (−4.2%)◦	.496 (−6.3%)▽	.379 (−3.3%)◦
Google	.436 (−6.8%)▽	.486 (−6.0%)◦	.491 (−7.1%)▽	.347 (−11.3%)▼
		C@5		
Both	.627	.676	.678	.535
Teoma	.595 (−5.1%)▽	.617 (−8.8%)▼	.634 (−6.5%)▼	.514 (−3.9%)◦
Google	.595 (−5.1%)◦	.607 (−10.1%)▼	.628 (−7.5%)▼	.488 (−8.7%)▼

(a) Individual test sets

	MRR	C@1	C@5
Both	.537	.477	.630
Teoma	.508 (−5.4%)▼	.454 (−4.9%)▼	.591 (−6.1%)▼
Google	.495 (−7.8%)▼	.441 (−7.6%)▼	.581 (−7.7%)▼

(b) Overall performance

this architecture takes advantage of existing infrastructure (and bypasses an enormous engineering challenge), it is a potential source of concern because Web search engines represent external components whose input/output characteristics are not well defined. Among other issues, the exact semantics of query operators are unclear, and the ranking algorithms are closely guarded secrets. This section examines the impact of different search engine parameters and concludes that data redundancy manifests in a performance curve that remains qualitatively invariant across a number of confounding factors. This stable characteristic allows factoid systems to rely on external Web search engines for basic retrieval functionality.

6.1 Using Different Search Engine

The current implementation of Aranea employs both Google and Teoma for retrieving snippets from the Web: this immediately suggests three experiments, to determine answer accuracy with one, the other, or both engines. These results are shown in Table IV. The default configuration of Aranea uses baseline queries as well as inexact and exact reformulations; up to 100 snippets per query are fetched. The only difference between the three runs is the source of the snippets.

Not surprisingly, the combination of both search engines outperforms either one individually—differences in all metrics are statistically significant at the 1% level when all four test sets are considered. Although it appears that Teoma outperforms Google when used individually, the differences in performance on all metrics are not statistically significant, with the exception of MRR on the TREC-12 test set (at the 5% level).

It is important to note that these results *do not* allow us to draw conclusions about the overall performance of individual search engines. Our experiments

used data from the Web at a particular point in time—the performance of individual search engines does not appear to be temporally stable, as we will discuss in Section 6.3. The only generalizable conclusion that can be drawn is that using two search engines is better than using only one, which is consistent with the central principle behind data redundancy (i.e., more is better). This finding is supported by previous work [Chu-Carroll et al. 2003].

6.2 Mining Different Numbers of Snippets

Implicit in the redundancy-based approach to factoid QA is the idea that "the more data, the better." This maxim appears to be supported by experiments described in the previous section. Indeed, the property of data redundancy is touted as the primary driver of performance. Extending the work of Dumais et al. [2002], we conducted a series of experiments to examine this claim in more detail. Results reveal that this catchy slogan is an oversimplification.

Is more always better? It is possible to address this question by varying the number of snippets presented to Aranea. In order to isolate the relevant experimental variable, runs were conducted only with baseline queries, on either Google or Teoma individually. No inexact or exact reformulations were used, since the number of snippets they returned was more variable and therefore harder to control. The only independent variable in these experiments was the number of text snippets fetched from either search engine. Graphs of the mean reciprocal rank, plotted against the number of snippets requested (on a logarithmic scale), are shown in Figure 6. Plots are shown for each individual test set, as well as across all available questions.

The results of this experiment show that although MRR increases with the number of snippets requested, little is gained beyond a certain point, and in fact, performance decreases slightly as more snippets are presented to Aranea. Table V quantifies this observation by comparing the peak MRR to MRR at 1000 snippets. Table VI shows the number of snippets that yields the peak MRR. Overall, the decrease in MRR from using too many snippets is statistically significant, which runs counter to the "more is better" claim.

At first glance, the shape of these graphs replicate exactly the findings reported in Dumais et al. [2002]. Both studies reveal that performance increases as more snippets are presented to the system, but only up to a certain point; slight drops in MRR are observed at the tail ends of the curves. For the AskMSR system, it's unclear if the performance drop is statistically significant because the proper tests were not performed.

Dumais et al. [2002], page 295, offered the following explanation for the shape of the curve:

> We believe that the slight drop at the high end is due to the increasing influence that the weaker rewrites have when many snippets are returned. The most restrictive rewrites return only a few matching documents. Increasing the number of snippets increases the number of the least restrictive matches (the AND matches), thus swamping the restrictive matches. In addition, frequent n-grams begin to dominate our rankings at this point.

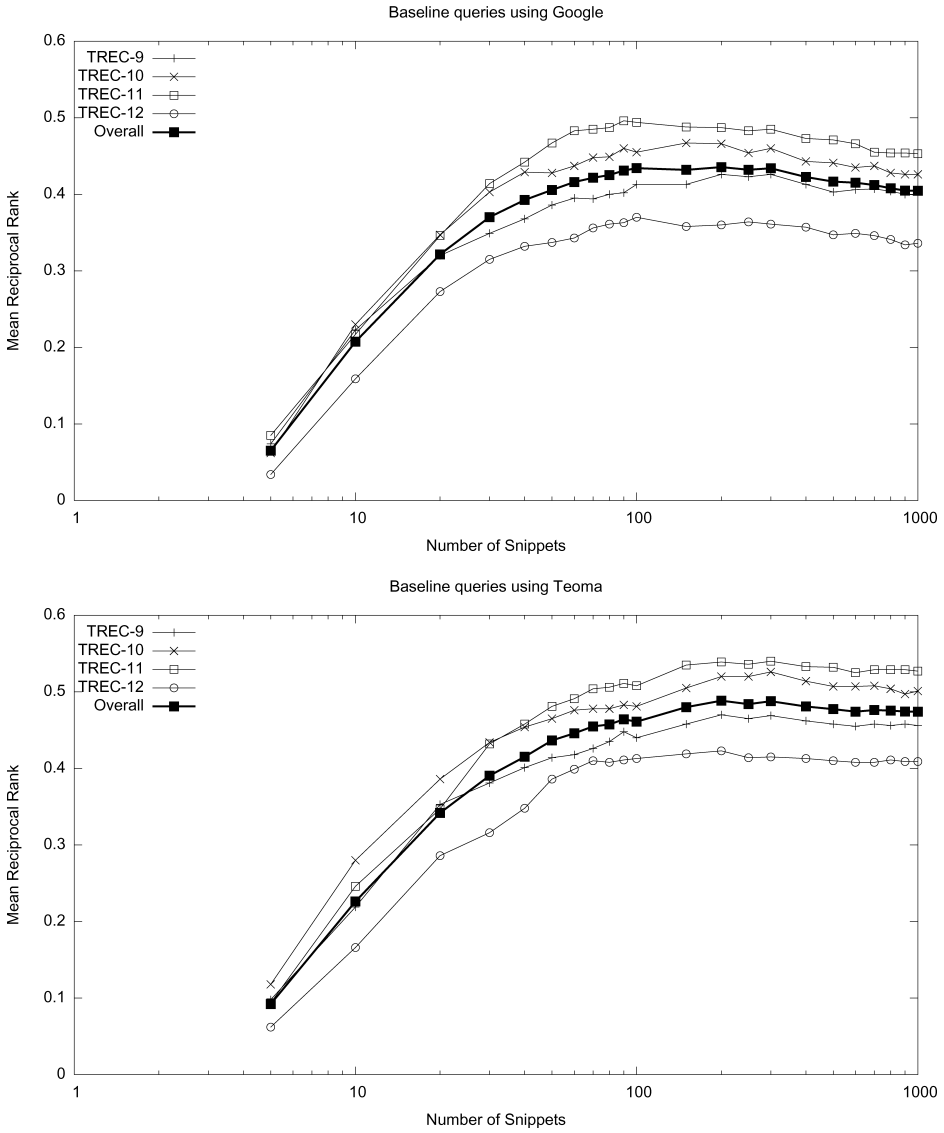

Fig. 6. MRR versus number of snippets for Google (top) and Teoma (bottom): baseline queries only.

Upon closer examination, it appears that their experiment conflated two factors: the impact of different query rewrites (reformulations) and the impact of snippet count. In contrast, our experiment focused solely on the second factor, since only baseline queries were used (experiments examining the effect of different reformulations will be described in Section 7.1). Thus, it appears that different query reformulations cannot be the sole cause for the shape of the MRR curves, given what we observe in Figure 6.

Nevertheless, two different studies on different systems arrive at the same basic conclusion. This finding suggests that Figure 6 depicts a general

Table V. Peak MRR versus MRR at 1000 Snippets Fetched for Google and Teoma

	TREC-9	TREC-10	TREC-11	TREC-12
MRR (Google)				
At peak	.426	.467	.496	.370
At 1k snippets	.399 (−6.3%)▼	.426 (−8.8%)▼	.453 (−8.7%)▼	.336 (−9.2%)▽
MRR (Teoma)				
At peak	.470	.526	.540	.423
At 1k snippets	.456 (−3.0%)▽	.501 (−4.8%)▼	.527 (−2.4%)▽	.409 (−3.3%)○

(a) Individual test sets

	Google	Teoma
At peak	.436	.488
At 1k snippets	.405 (−7.1%)▼	.474 (−3.0%)▼

(b) Overall performance

Table VI. Number of Snippets That Yields the Peak MRR for Google and Teoma

	TREC-9	TREC-10	TREC-11	TREC-12	Overall
Google	200	150	90	100	200
Teoma	200	300	300	200	200

characteristic of the Web, and is not the result of idiosyncrasies that can be attributed to individual search engines or particular implementations of redundancy-based techniques. We further examine this issue below.

A side effect of having little control over external components (Google and Teoma) is that Aranea can only *request* a certain number of snippets for each query; the actual number of snippets retrieved is often different. Although search engines claim a large number of hits in response to a query, there is usually an upper bound on the number of snippets that can be gathered by simulating a click on the "next page" link in the result set (which is how Aranea operates; see Section 4.1). This appears to be an unavoidable limitation of Google and Teoma at the time of these experiments, and reinforces the need to better understand question answering performance given idiosyncratic behaviors of external components. In truth, the x axes of the graphs shown in Figure 6 specify the number of snippets requested by Aranea, *not* the actual number of text snippets returned by either Google or Teoma.

How often are fewer snippets returned than requested? Figure 7 provides an answer: it plots the fraction of questions in each test set that fetches the requested number of snippets. As can be seen, Google doesn't return *any* hits for approximately 15% of the questions, while that number appears to be smaller for Teoma.[7] The shapes of the two graphs highlight the differences in behavior between the two search engines.

[7]This was the observed behavior of Google in September 2005, when these experiments were conducted. At the time, it appeared Google employed an exact-match algorithm that attempts to retrieve pages with *all* query terms—as a result, many long factoid questions retrieved zero hits. As of June 2006, Google's algorithm appeared to implement a best-match scheme—the result sets indicate which query terms were dropped.

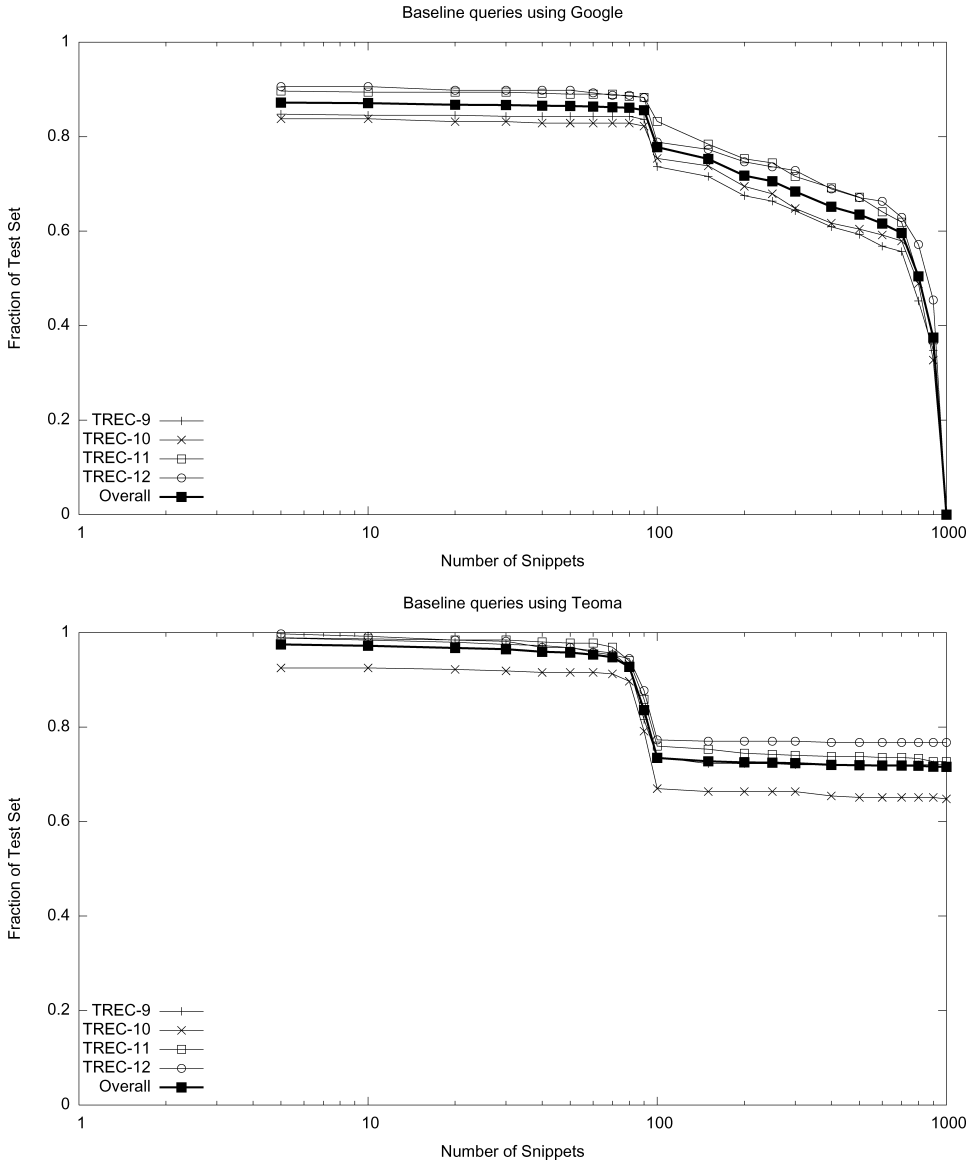

Fig. 7. Fraction of questions in each test set that fetches the requested number of snippets; Google (top) and Teoma (bottom).

Does this "not enough snippets" phenomenon affect the shape of the curves shown in Figure 6? To answer this, we created a reduced test set containing only questions that fetched at least 900 snippets. For the experiments involving Google, this filtering process yielded 153 questions for TREC-9, 105 for TREC-10, 168 for TREC-11, and 174 for TREC-12 (600 total). For the experiments involving Teoma, the numbers were 317, 208, 330, and 294, for each of the four test sets, respectively (1149 total). The experiment whose results are shown

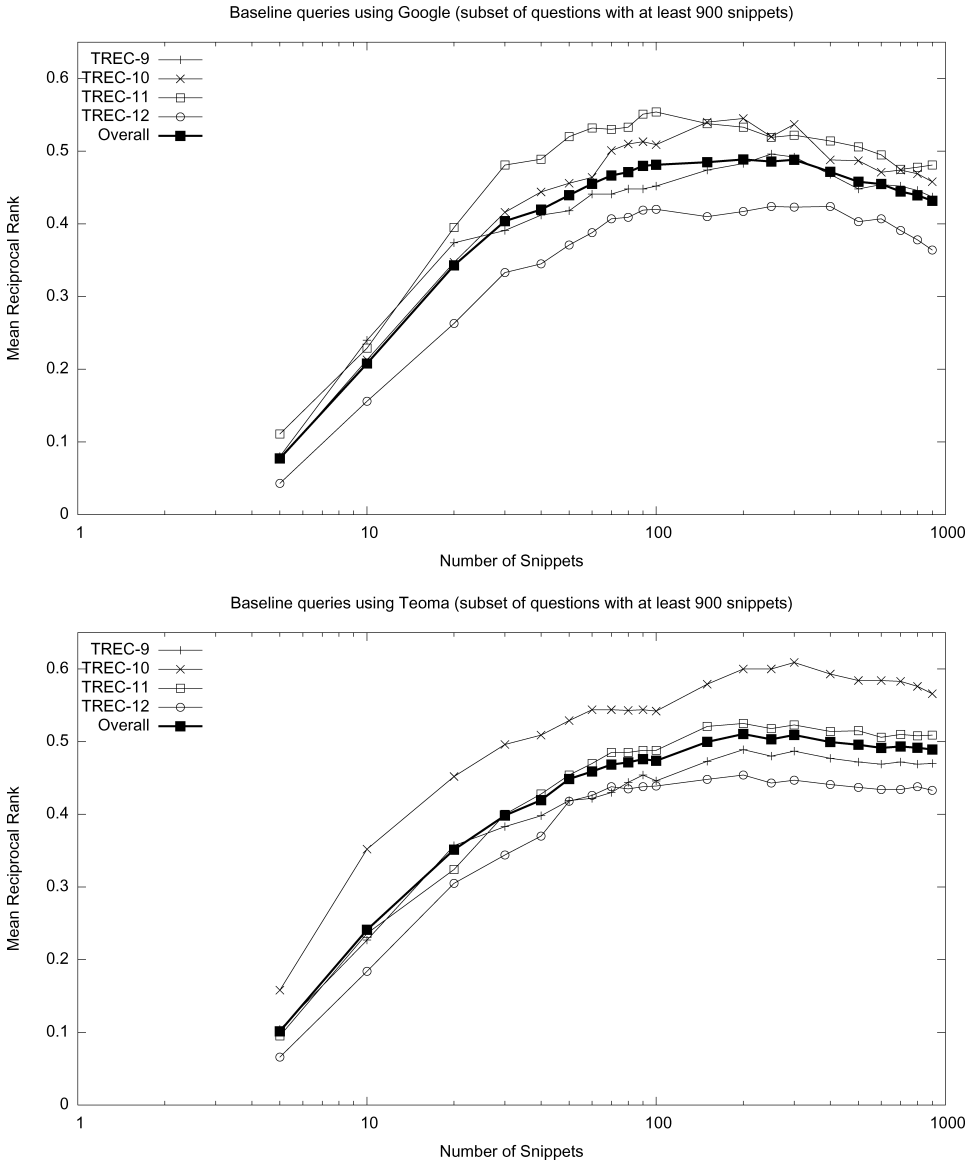

Fig. 8. MRR versus number of snippets for Google (top) and Teoma (bottom) on questions that fetched at least 900 snippets: baseline queries only.

in Figure 6 was replicated with these reduced test sets (i.e., the independent variable was the number of snippets retrieved using baseline queries; no reformulations were used). The results are shown in Figure 8. The shapes of the curves are the same, although the graph for Google appears a bit noisy because there are fewer questions.

We believe that the shape of the curves seen in Figures 6 and 8 can be attributed to the decreasing quality of snippets as a redundancy-based system

fetches more and more results. It is a reasonable assumption that Web search engines obey the *probabilistic ranking principle* [Robertson 1977], which simply states that systems should rank results based on how likely they are to satisfy users' information needs, that is, relevant documents should be ranked before irrelevant documents. Common experience with Web search engines confirms this assumption. Thus, redundancy-based techniques must strike a balance between two competing factors: fewer snippets from higher-quality documents (i.e., those more likely to be relevant), or more snippets from lower-quality documents (i.e., those less likely to be relevant). In the first case, the danger lies in not having enough data redundancy (i.e., diversity of answer forms); in the second case, the danger lies in data redundancy reinforcing the wrong answer (i.e., spurious high-frequency n-grams).

To put it differently, data redundancy can be exploited by observing statistical associations between query terms and candidate answers: on the one hand, a system needs sufficient data to observe these associations, but on the other hand, such associations do not necessarily lead to answers, particularly in lower-quality snippet text. On this particular set of TREC questions, the optimal point that balances these competing factors appears to be around 200 snippets. Note that this explanation is consistent with the reasoning provided by Dumais et al. [2002]: more accurate rewrites (which are placed before backoff queries in the AskMSR system) provide higher-quality snippets, and beyond a certain point, information in lower-quality snippets will tend to dominate.

The experiments described thus far examine the effect of snippet quantity in the context of a single search engine. What happens when a system combines results from multiple search engines? Although the graphs in Figure 6 indicate that increasing the number of snippets from 100 to 200 has little effect on MRR, these are snippets from the *same search engine*. Alternatively, pulling in snippets from *another search engine* does have a statistically significant impact on answer accuracy, as shown in Table IV. There appears to be an interesting interaction here, which is further explored below.

The key issue can be boiled down to comparing QA performance under three different conditions: fetching n snippets from Google, n snippets from Teoma, or a total of n snippets from both (i.e., half from each). The results of these experiments are shown in Figure 9, with various values of n on the x axis. Only baseline queries were employed as to not conflate the effects of different reformulations. Each graph contrasts Google snippets only, Teoma snippets only, and Google+Teoma snippets. Note that these experiments were performed on the full set of TREC questions and hence contains questions for which fewer than the requested number of snippets were fetched. But as we demonstrated, this should not have a qualitative impact on performance. The graphs show that, for almost any number of snippets, using results from both search engines yields higher MRR than either alone.

What is the overlap between snippets returned by Google and Teoma? This is an important question, since redundant information returned by both search engines is a confounding factor in these experiments. At the exact snippet level, we discovered no overlap, since the search engines employ different techniques

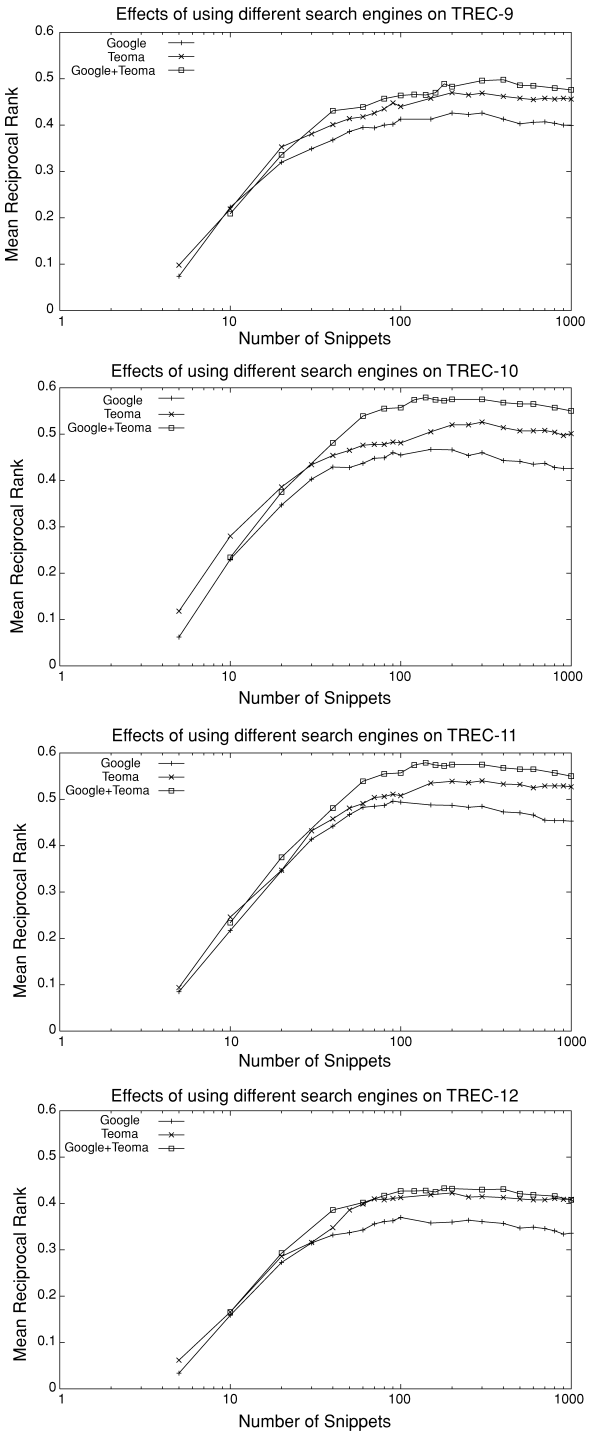

Fig. 9. MRR versus number of snippets, comparing taking snippets from exclusively from Google, exclusively from Teoma, or half from each: baseline queries only.

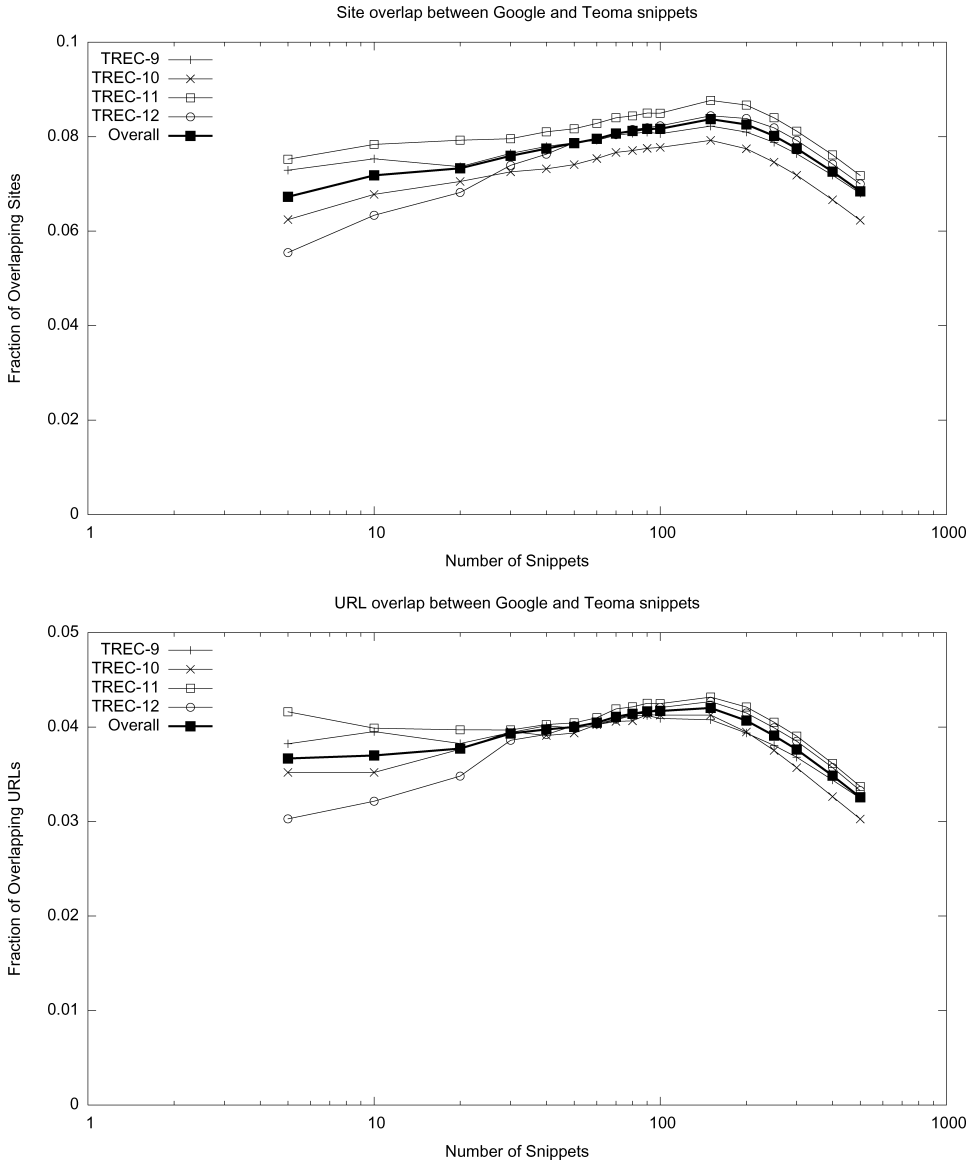

Fig. 10. Overlap between Google and Teoma results, shown as a function of the number of total snippets requested: mean of prequestion overlap in terms of Web sites (top) and in terms of complete URLs (bottom).

for generating contextual summaries of the underlying pages (so that the snippets are different even if they come from the same URL). However, it is possible to compute overlap at the site and URL levels. No effort was made to normalize domain names possibly belonging to the same organization, so that `www.umd.edu` and `www.umiacs.umd.edu`, for example, would be considered different. These results are shown in Figure 10. Each data point represents the mean

of per-question fractional overlap, that is, the number of overlapping sites or URLs divided by the total number of unique sites or URLs. Higher variance in the fractional overlap on a per-question basis is observed for smaller numbers of snippets. Across all values of n, the mean overlap in sites is less than 10%, and less than 5% when unique URLs are considered.

What's the difference between using more snippets from the same search engine and using snippets from a different search engine? It appears that, in both cases, new data is brought in—overlapping URLs account for only a small fraction of the newly acquired results from a different search engine, and even if the same URLs were retrieved, the generated snippets would still be different. We believe that the difference lies in the *quality* of the snippets returned. When fetching results from multiple search engines, Aranea is harvesting snippets that are more highly ranked, and hence higher in quality (i.e., more likely to be relevant). This conclusion is consistent with experiments on individual search engines (e.g., results shown in Figure 6). Redundancy-based techniques need to balance quality with quantity, and using multiple search engines provides alternative sources for quality.

To summarize, if the principle of data redundancy translates into the slogan "the more data, the better," a closer examination would call for a refinement: "the more data, the better, but not too much garbage." In contrast to the data-driven philosophy espoused in Banko and Brill [2001], simply throwing more data at the problem of factoid question answering doesn't help. Data redundancy can be leveraged to increase performance, but only up to a certain point. Experimental results suggest that the shape of this performance curve represents an inherent property of the Web. This finding appears to be a valid generalization, and not an Aranea-specific idiosyncrasy, since the same trends were observed in multiple experiments under very different conditions (i.e., different experimental setups, different system implementations, different search engines, questions that return different numbers of snippets, etc.).

The predictable behavior of Web search engines under a variety of conditions is an important finding. This result allows one to build factoid QA systems that rely on these external components, hence bypassing the enormous engineering challenge of crawling and indexing the entire Web.

6.3 The Changing Web

As the Web grows and evolves, the coverage of search engines changes as well; see, for example, Bar-Ilan [2002] and Ntoulas et al. [2004]. It would be interesting to examine question answering performance as a function of time. Because Aranea implements a caching mechanism for Web pages, it is possible to "turn back the clock" and run experiments on any set of previously-acquired snippet data. Table VII shows runs on the TREC-9 and TREC-10 test sets using data gathered from September 2005 and June 2004. The results employ the default configuration of Aranea that uses all query types. We experimented with Google alone, with Teoma alone, and with both search engines. The overlap between 2004 and 2005 data is shown in Table VIII. Figures are broken down according to search engine combination (Google+Teoma, Google, Teoma). As before, we

Table VII. Historical Runs from 2004 and 2005: All Queries

	TREC-9	TREC-10	Overall
MRR			
Google+Teoma (2005)	.530	.582	.552
Google+Teoma (2004)	.521 (−1.7%)$^{\circ}$	.609 (+4.6%)$^{\circ}$	.558 (+1.1%)$^{\circ}$
Google (2005)	.497	.532	.512
Google (2004)	.468 (−5.8%)$^{\circ}$	.528 (−0.8%)$^{\circ}$	.493 (−3.6%)$^{\circ}$
Teoma (2005)	.502	.545	.520
Teoma (2004)	.496 (−1.2%)$^{\circ}$	.581 (+6.6%)$^{\triangle}$	.532 (+2.3%)$^{\circ}$
C@1			
Google+Teoma (2005)	.468	.517	.489
Google+Teoma (2005)	.464 (−1.0%)$^{\circ}$	.548 (+6.0%)$^{\circ}$	.500 (+2.2%)$^{\circ}$
Google (2005)	.436	.486	.457
Google (2004)	.411 (−5.7%)$^{\circ}$	.470 (−3.2%)$^{\circ}$	.436 (−4.6%)$^{\circ}$
Teoma (2005)	.445	.495	.466
Teoma (2004)	.434 (−2.6%)$^{\circ}$	.523 (+5.7%)$^{\circ}$	.472 (+1.1%)$^{\circ}$
C@5			
Google+Teoma (2005)	.627	.676	.648
Google+Teoma (2005)	.616 (−1.8%)$^{\circ}$	.698 (+3.2%)$^{\circ}$	.650 (+0.4%)$^{\circ}$
Google (2005)	.595	.607	.601
Google (2004)	.552 (−7.3%)$^{\triangledown}$	.617 (+1.5%)$^{\circ}$	.580 (−3.5%)$^{\circ}$
Teoma (2005)	.595	.617	.604
Teoma (2004)	.593 (−0.4%)$^{\circ}$	.667 (+8.1%)$^{\blacktriangle}$	.624 (+3.3%)$^{\circ}$

Table VIII. Snippet Overlap between 2004 and 2005 Data

	TREC-9	TREC-10	Overall
URL overlap			
Google+Teoma	13.9%	13.6%	13.8%
Google	9.8%	10.3%	10.0%
Teoma	16.2%	15.0%	15.7%
Site overlap			
Google+Teoma	19.8%	19.2%	19.5%
Google	16.7%	16.7%	16.7%
Teoma	20.0%	18.8%	19.5%

computed two separate measures of fractional overlap (intersection divided by union): in terms of unique URLs and in terms of unique Web sites.

Although question answering performance fluctuates, in most cases the differences in performance are not statistically significant. In general, results from individual search engines vary more than results from employing both—it appears that mining snippets from multiple sources "evens out" the effects of changes in Google and Teoma data. Interestingly, the overlap in results from the 2 years is low (both in terms of unique URLs and unique sites). It appears that Aranea is essentially working with different pages. Yet, we note that the content of the snippets, at least at the n-gram level, remains similar. Although we cannot draw general conclusions about the temporal evolution of the Web based on this limited data, our experiments show that Aranea's performance remained stable over at least this observed time frame.

7. THE ROLE OF KNOWLEDGE IN DATA REDUNDANCY

A rough characterization of the redundancy-based approach to factoid question answering would be "data-driven," while the traditional approach might be described as "knowledge-driven," given its reliance on ontological resources and fine-grained detection of semantic entities. However, as we have begun to show in the previous section, the quantity of data is not the sole factor affecting system performance.

In this section, we examine the role of heuristic knowledge in the redundancy-based approach. While such knowledge is very different from the formal ontological resources deployed in more traditional QA systems, it nevertheless encodes regularities, patterns, and generalizations about factoid questions and their relationships to answers. In this section, we specifically discuss two places in the question answering pipeline where knowledge is brought to bear: in the question analysis and answer extraction phases.

7.1 Question Analysis

Aranea issues three types of queries to fetch snippets from the Web: baseline, inexact reformulations, and exact reformulations. To summarize (see Section 4.1 for more details):

—The baseline query is simply the original factoid question verbatim.
—Exact reformulations anticipate the location of the answer, for example, the **?x** in "**?x** shot Abraham Lincoln" indicates where the system expects to find the answer to "Who shot Abraham Lincoln?"
—Inexact reformulations utilize query terms from the reformulated queries (as a bag of words), but do not attempt to match unbound variables.

In some cases, inexact reformulations differ from the original baseline queries only in the order of the query terms and the morphology of the verbs. However, this in practice does produce different results because both Google and Teoma are sensitive to the morphology of query terms and their order. In other cases, inexact reformulations contain terms not found in the original question, which yields a query expansion effect, for example, "the Valley of the Kings is located in" for the question "Where is the Valley of the Kings?"

Aranea's three query types map into two different approaches for leveraging data redundancy: exploiting the statistical associations between question and answer terms (baseline and inexact reformulations), and directly extracting answers using surface patterns (exact reformulations).

Exact and inexact reformulations implicitly encode knowledge about the structure of language (in general) and regularities in factoid questions (in particular). The regular expression patterns that translate input questions into fragments of declarative sentences are in fact performing question analysis. They encode, albeit in a very crude manner, knowledge about the position of arguments and verbs in *wh*-questions and linguistic phenomena such as *wh*-movement. Since a large number of factoid questions exhibit the same syntactic structure (see Section 5.1), a relatively small number of pattern matching rules appears sufficient to capture a significant amount of regularity.

Table IX. Impact of Different Query Types on Question Answering Accuracy: Both Google and Teoma Snippets. Key: Baseline (b), Inexact (i), and Exact (e)

	TREC-9	TREC-10	TREC-11	TREC-12
		MRR		
b+i+e	.530	.582	.587	.448
b+i	.501 (−5.5%)▼	.571 (−1.9%)◦	.584 (−0.5%)◦	.437 (−2.5%)◦
b+e	.519 (−2.1%)◦	.559 (−4.0%)▽	.591 (+0.7%)◦	.451 (+0.7%)◦
i+e	.362 (−31.7%)▼	.400 (−3.1%)▼	.406 (−30.8%)▼	.203 (−54.7%)▼
b	.481 (−9.2%)▼	.531 (−8.8%)▼	.575 (−2.0%)◦	.432 (−3.6%)◦
e	.174 (−67.2%)▼	.168 (−71.1%)▼	.150 (−74.4%)▼	.100 (−77.7%)▼
i	.284 (−46.4%)▼	.330 (−43.3%)▼	.347 (−40.9%)▼	.157 (−65.0%)▼
		C@1		
b+i+e	.468	.517	.529	.392
b+i	.434 (−7.3%)▼	.502 (−3.0%)◦	.526 (−0.4%)◦	.376 (−4.0%)◦
b+e	.457 (−2.4%)◦	.498 (−3.6%)◦	.533 (+0.8%)◦	.399 (+2.0%)◦
i+e	.318 (−32.0%)▼	.371 (−28.3%)◦	.361 (−31.7%)▼	.175 (−55.3%)▼
b	.416 (−11.2%)▼	.458 (−11.4%)▼	.518 (−2.1%)◦	.381 (−2.7%)◦
e	.159 (−66.0%)▼	.162 (−68.7%)▼	.137 (−74.2%)▼	.094 (−76.0%)▼
i	.245 (−47.6%)▼	.299 (−42.2%)▼	.304 (−42.5%)▼	.128 (−67.3%)▼
		C@5		
b+i+e	.627	.676	.678	.535
b+i	.602 (−4.0%)▼	.676 (−0.0%)◦	.674 (−0.6%)◦	.533 (−0.5%)◦
b+e	.618 (−1.4%)◦	.654 (−3.2%)▽	.670 (−1.3%)◦	.527 (−1.5%)◦
i+e	.432 (−31.2%)▼	.445 (−34.1%)▼	.469 (−30.8%)▼	.243 (−54.6%)▼
b	.582 (−7.2%)▼	.648 (−4.1%)◦	.656 (−3.2%)▽	.512 (−4.4%)▽
e	.198 (−68.5%)▼	.178 (−73.7%)▼	.167 (−75.3%)▼	.110 (−79.5%)▼
i	.348 (−44.6%)▼	.374 (−44.7%)▼	.412 (−39.3%)▼	.204 (−62.0%)▼

(a) Individual test sets

	MRR	C@1	C@5
b+i+e	.537	.477	.630
b+i	.523 (−2.5%)▼	.460 (−3.5%)▼	.621 (−1.4%)▽
b+e	.531 (−1.1%)◦	.473 (−0.8%)◦	.618 (−1.8%)▽
i+e	.344 (−35.9%)▼	.307 (−35.7%)▼	.400 (−36.5%)▼
b	.506 (−5.8%)▼	.445 (−6.7%)▼	.600 (−4.8%)▼
e	.148 (−72.4%)▼	.138 (−71.1%)▼	.164 (−74.0%)▼
i	.281 (−47.7%)▼	.245 (−48.7%)▼	.337 (−46.5%)▼

(b) Overall performance

Therefore, understanding the relative contributions of these different query types is a step toward understanding the impact of knowledge in redundancy-based techniques. We conducted a series of experiments that examined the performance of each query type individually and in combination. The goal was to determine how much performance suffers when knowledge encoded in the query reformulation rules are ablated. Results are shown in Table IX. Each row heading encodes the types of queries used: **b**aseline, **i**nexact, and **e**xact. Otherwise, the default configuration of Aranea was used, which employs both Google and Teoma and retrieves up to 100 snippets per query. All runs were compared to the condition that uses all three query types (run b+i+e).

Overall, the results show that removing inexact queries yields in a small drop in MRR that is not statistically significant (b+i+e vs. b+e), while removing exact

Table X. Using All Queries Versus Using Baseline Queries Only: Google Snippets Only

	TREC-9	TREC-10	TREC-11	TREC-12
		MRR		
b+i+e	.497	.532	.547	.400
b	.413 (−16.9%)▼	.455 (−14.5%)▼	.494 (−9.7%)▼	.370 (−7.5%)▼
		C@1		
b+i+e	.436	.486	.491	.347
b	.357 (−18.2%)▼	.396 (−18.6%)▼	.441 (−10.3%)▼	.316 (−9.0%)▼
		C@5		
b+i+e	.595	.607	.628	.488
b	.502 (−15.6%)▼	.551 (−9.2%)◦	.568 (−9.5%)▼	.457 (−6.4%)▽

(a) Individual test sets

	MRR	C@1	C@5
b+i+e	.495	.441	.581
b	.434 (−12.3%)▼	.379 (−14.1%)▼	.520 (−10.5%)▼

(b) Overall performance

queries significantly reduces MRR (b+i+e vs. b+i). Exact and inexact reformulations do not appear to be effective by themselves, since rules for generating them are low in coverage and do not apply to many questions (more on this below). However, removing both exact and inexact reformulations hurts MRR by only 5.8% (b+i+e vs. b), which suggests that the linguistic knowledge encoded in the reformulation rules do not contribute a great deal to answer accuracy. It seems that direct attempts to apply surface patterns for answer extraction have a relatively minor impact on performance beyond what can already be achieved with baseline queries.

These results appear inconsistent with those reported by Dumais et al. [2002]. On the TREC-9 test set, the AskMSR system achieved an MRR of .507 using all query rewrites and .450 using the baseline queries only, a drop of 11.2%. However, the AskMSR results employed Google snippets only, whereas the configuration of Aranea used here employed both Google and Teoma snippets. To isolate the impact of query reformulations, we conducted two separate sets of experiments as a followup: one using Google snippets only, the other using Teoma snippets only. Both sets of experiments contrasted using all three query types (b+i+e) versus using only the baseline query (b).

Table X shows the impact of query reformulations, with respect to Google snippets only; Table XI shows the same, with respect to Teoma snippets only. We note that there are much larger differences in performance between using all queries and using baseline queries only. These results reveal an interesting interaction between data redundancy (pulling in snippets from multiple sources) and linguistic knowledge (as captured in the reformulation rules). When using only one search engine (either Google or Teoma), query reformulations matter a great deal, which is consistent with previous work. However, the impact of query reformulations is lessened when a system draws snippets from multiple sources, suggesting that whatever is lost in more precise queries can be mostly recovered by simply providing the system with more high-quality snippets.

Table XI. Using All Queries Versus Using Baseline Queries Only: Teoma Snippets Only

	TREC-9	TREC-10	TREC-11	TREC-12
		MRR		
b+i+e	.502	.545	.550	.434
b	.440 (−12.4%)▼	.482 (−11.6%)▼	.508 (−7.6%)▼	.413 (−4.8%)▼
		C@1		
b+i+e	.445	.495	.496	.379
b	.377 (−15.3%)▼	.424 (−14.5%)▼	.449 (−9.3%)▼	.368 (−2.8%)▼
		C@5		
b+i+e	.595	.617	.634	.514
b	.541 (−9.2%)▼	.558 (−9.6%)◦	.595 (−6.3%)▼	.483 (−6.1%)▽

(a) Individual test sets

	MRR	C@1	C@5
b+i+e	.508	.454	.591
b	.461 (−9.2%)▼	.405 (−10.8%)▼	.546 (−7.7%)▼

(b) Overall performance

Nevertheless, query reformulations still provide some performance gain, even when both Google and Teoma snippets are used.

Although it would be interesting to compare AskMSR's rewrite rules and Aranea's query reformulation rules, such a comparison is unfortunately not possible since the author does not have access to the original AskMSR source code. However, based on available system descriptions and recollections of the author, the question analysis components of both systems are comparable. Both employ rules that operate on surface strings; neither system performs any linguistic analysis other than part-of-speech tagging. Aranea's rules are less prone to overgeneration, primarily because some rules in the AskMSR system purposely overgenerate. For example, to undo *aux*-movement in a question like "Where is the Louvre located?" the system exhaustively generates reformulations with the auxiliary verb in every possible surface position; see Brill et al. [2001] and Dumais et al. [2002]. Naturally, most of these positions yield invalid patterns, but this was not a concern because they failed to fetch any snippets from the Web. Based on the limited results available (on TREC-9), Aranea's reformulations and AskMSR's rewrite rules appear to be roughly comparable in terms of performance. These results are also consistent with the work of Agichtein et al. [2004], who demonstrated that it is possible to learn reformulations specific to individual search engines that improve factoid question answering performance.

Although Table IX shows that exact reformulations alone are ineffective for answering factoid questions, one should not readily dismiss the role of surface pattern matching. Could it be the case that exact reformulations represent a different tradeoff between accuracy and coverage? To illustrate, compare "Who is the Greek God of the Sea?" with "What was the name of the famous battle in 1836 between Texas and Mexico?" Whereas the exact pattern "**?x** is the Greek God of the Sea" fetches plenty of matches from the Web, the exact pattern "**?x** is the name of the famous battle in 1836 between Texas and Mexico" yields

Table XII. Impact of Different Query Types on Question Answering Accuracy (on the Subset of Questions for Which Exact Reformulations Produced an Answer): Both Google and Teoma Snippets

	TREC-9	TREC-10	TREC-11	TREC-12
# Q's	114	72	106	67
		MRR		
e	.673	.750	.642	.569
b	.579 (−14.0%)$^{\circ}$	.666 (−11.2%)$^{\circ}$	.676 (+5.3%)$^{\circ}$	.494 (−13.2%)$^{\circ}$
b+i	.620 (−7.9%)$^{\circ}$	.754 (+0.5%)$^{\circ}$	.718 (+11.8%)$^{\circ}$	.516 (−9.3%)$^{\circ}$
b+i+e	.732 (+8.8%)$^{\circ}$	.807 (+7.6%)$^{\circ}$	.725 (+12.9%)$^{\blacktriangle}$	.552 (−3.0%)$^{\circ}$
		C@1		
e	.614	.722	0.585	.537
b	.500 (−18.6%)$^{\triangledown}$	.597 (−17.3%)$^{\triangledown}$	.613 (+4.8%)$^{\circ}$	.448 (−16.7%)$^{\circ}$
b+i	.535 (−12.9%)$^{\circ}$	.722 (+0.0%)$^{\circ}$	.670 (+14.5%)$^{\circ}$	.448 (−16.7%)$^{\circ}$
b+i+e	.667 (+8.6%)$^{\circ}$	.792 (+9.6%)$^{\circ}$	.670 (+14.5%)$^{\triangle}$	.507 (−5.6%)$^{\circ}$
		C@5		
e	.763	.792	.717	.627
b	.702 (−8.0%)$^{\circ}$	.792 (+0.0%)$^{\circ}$	.764 (+6.6%)$^{\circ}$	.567 (−9.5%)$^{\circ}$
b+i	.737 (−3.4%)$^{\circ}$	.819 (+3.5%)$^{\circ}$	.792 (+10.5%)$^{\circ}$	.612 (−2.4%)$^{\circ}$
b+i+e	.833 (+9.2%)$^{\circ}$	.833 (+5.3%)$^{\circ}$	.802 (+11.8%)$^{\triangle}$	.612 (−2.4%)$^{\circ}$

(a) Individual test sets

	MRR	C@1	C@5
e	.660	.613	.730
b	.609 (−7.7%)$^{\circ}$	.543 (−11.4%)$^{\triangledown}$	.713 (−2.3%)$^{\circ}$
b+i	.656 (−0.5%)$^{\circ}$	.596 (−2.7%)$^{\circ}$	.747 (+2.3%)$^{\circ}$
b+i+e	.711 (+7.8%)$^{\blacktriangle}$	.663 (+8.2%)$^{\triangle}$	.783 (+7.3%)$^{\blacktriangle}$

(b) Overall performance (359 questions)

far fewer (if any) results. In the latter case, Aranea would simply return "don't know," which is arguably better than returning the wrong answer. Could it be the case that while exact reformulations have limited coverage, they are highly accurate in the cases where they do generate answers?

To test this hypothesis, questions which returned "don't know" using only exact reformulations were discarded, producing four smaller test sets. The size of each test set and answer accuracy under different experimental conditions are shown in Table XII (snippets from both Google and Teoma were used). Given the hypothesis being explored, the table presents relative differences with respect to the Aranea variant that utilizes exact reformulations only (e).

These results do indeed show that using exact reformulations beats using baseline queries, but the results are not statistically significant for MRR and only significant at the 5% level for C@1. Adding inexact reformulations to baseline queries raises answer accuracy on all metrics to about the same level as using exact reformulations alone. It appears that, for certain questions, exact reformulations by themselves work well, but using all three query types still yields statistically significant increases in accuracy. This finding is consistent with the experimental results shown in Table IX.

To round out our understanding of Aranea's question analysis component, we experimented with weights assigned to different query types. In Aranea,

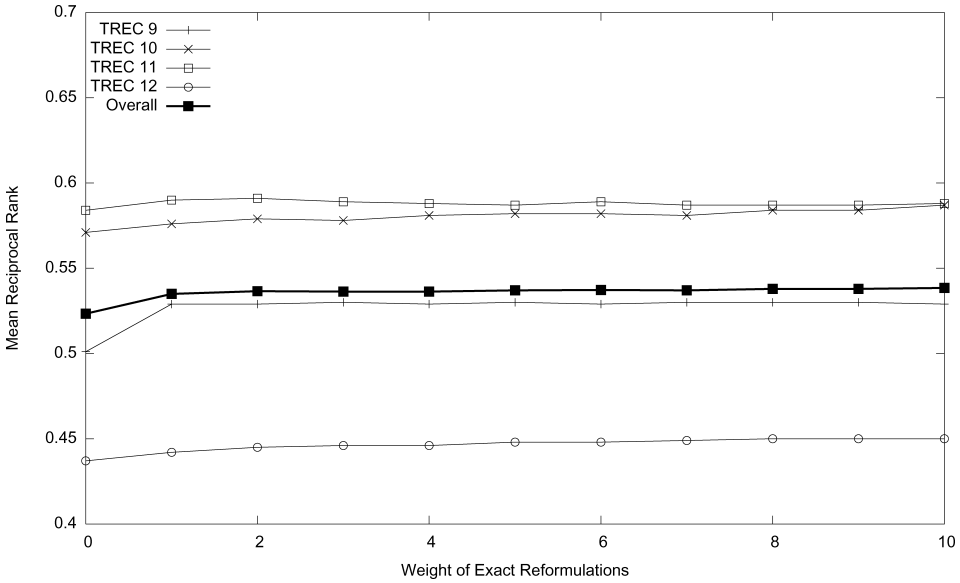

Fig. 11. MRR versus weight of exact reformulations.

n-grams generated from exact reformulations are given a weight five times those generated from inexact reformulations and baseline queries. This parameter was arbitrarily set in the system development process, and the implications were never fully explored. Figure 11 plots the changes in MRR with varying values of this weight (using both Google and Teoma snippets). The results show that this parameter has little effect on question answering accuracy, beyond the introduction of exact reformulations themselves (weight of zero is the same as the b+i run in Table IX).

The experiments described in this section examine the question analysis component of Aranea—in particular, we assessed the impact of pattern-matching rules that generate inexact and exact reformulations. Since these rules implicitly encode linguistic knowledge about question and answer patterns, they provide an opportunity for us to examine the role of knowledge in redundancy-based question answering. Although reformulations have a substantial impact on accuracy when snippets are mined from a single Web search engine, the relative advantages they confer are reduced when snippets are gathered from multiple sources. This reveals an interesting interaction between having *more* data and having *better* data. This finding, to our knowledge, has not been reported before.

7.2 Answer Extraction

In the answer extraction phase of processing (see Figure 2), Aranea leverages data redundancy to mine answers from text snippets gathered from Web search engines. The analogous component in a traditional question answering architecture relies on named-entity recognizers to detect candidate answers of the correct semantic type, as determined by the question analysis module. This entire

process is supported by elaborate ontological resources, as previously discussed in Section 2.1. Surely this process can be characterized as "knowledge-driven." With redundancy-based methods, on the other hand, is there any "knowledge" to be found in the answer extraction phase? The answer is *yes*, as this section illustrates in greater detail.

Recall from Section 4.4 that Aranea implements three types of filters. In summary, they are as follows:

—*type-neutral*: throw away candidates that have query terms and that begin or end with stopwords;
—*type-specific*: for example, "how fast" answers must have a numeric component, "who" and "what" answers must be proper;
—*closed-class*: filter using a total of 17 types of fixed lists—for countries, languages, etc.

The second and third types encode associations between questions and answers. Type-specific filters capture broad generalizations about answer forms, whereas closed-class filters encode knowledge about a small set of answer types. Although both types of filters lack the formal structures associated with the ontologies used in traditional question answering systems (e.g., hierarchical structure that captures facts like "a country is a location"), they nevertheless encode "knowledge."

Following the theme of this section, we conducted a series of ablation studies that removed different filtering components. This allowed us to characterize the impact of these heuristics on question answering performance. The following experimental conditions were examined:

—*F0*: no filtering of any sort;
—*F1*: type-neutral filters only;
—*F2*: type-neutral and type-specific filters.

These runs were compared to the default Aranea configuration, which employed all three filter types. All runs used snippets from both Google and Teoma, and also took advantage of all query types (baseline, inexact, exact).

The results are shown in Table XIII. The overall differences in answer accuracy between any of the variants (F0, F1, F2) and the default condition are statistically significant at the 1% level across all metrics. Not surprisingly, of all filtering heuristics, the impact of type-neutral filters was the greatest. We see that type-specific filters also contributed a great deal to answer accuracy. The two types of filters appeared to be the centred driver in answer extraction. The closed-class filters have the smallest impact due to their limited applicability, but the effects are still noticeable and statistically significant.

Finally, what are the effects of the reranking heuristics? Whereas the various filters aim to discard wrong answers, the reranking module attempts to promote correct answers. Table XIII also shows the effect of removing the reranking module (indicated as run −Rerank): the result is a small but statistically significant drop in MRR and C@1, but no change in C@5 (which makes sense since reranking does not create any new answers).

Table XIII. Impact of Various Answer Extraction Components on Question Answering Accuracy: Google and Teoma Snippets, all Query Types

	TREC-9	TREC-10	TREC-11	TREC-12
		MRR		
Default	.530	.582	.587	.448
F0	.157 (−70.4%)▼	.213 (−63.4%)▼	.210 (−64.2%)▼	.129 (−71.2%)▼
F1	.449 (−15.3%)▼	.520 (−10.7%)▼	.480 (−18.2%)▼	.369 (−17.6%)▼
F2	.502 (−5.3%)▼	.560 (−3.8%)▽	.552 (−6.0%)▼	.419 (−6.5%)▼
−rerank	.518 (−2.3%)▽	.571 (−1.9%)▽	.574 (−2.2%)▽	.449 (−2.2%)▽
		C@1		
Default	.468	.517	.529	.392
F0	.098 (−79.1%)▼	.143 (−72.3%)▼	.150 (−71.7%)▼	.094 (−76.0%)▼
F1	.382 (−18.4%)▼	.442 (−14.5%)▼	.416 (−21.3%)▼	.300 (−23.3%)▼
F2	.436 (−6.8%)▼	.495 (−4.2%)◦	.491 (−7.1%)▼	.353 (−10.0%)▼
−Rerank	.452 (−3.4%)◦	.502 (−3.0%)◦	.509 (−3.8%)▽	.386 (−1.3%)◦
		C@5		
Default	.627	.676	.678	.535
F0	.257 (−59.1%)▼	.324 (−52.1%)▼	.315 (−53.6%)▼	.188 (−64.9%)▼
F1	.564 (−10.1%)▼	.642 (−5.1%)▼	.586 (−13.6%)▼	.488 (−8.8%)▼
F2	.609 (−2.9%)▽	.660 (−2.3%)◦	.654 (−3.6%)▽	.530 (−1.0%)◦
−Rerank	.625 (−3.6%)◦	.673 (−0.5%)◦	.674 (−0.6%)◦	.546 (−2.0%)◦

(a) Individual test sets

	MRR	C@1	C@5
Default	.537	.477	.630
F0	.177 (−67.1%)▼	.121 (−74.7%)▼	.270 (−57.1%)▼
F1	.453 (−15.7%)▼	.384 (−19.4%)▼	.568 (−9.8%)▼
F2	.508 (−5.4%)▼	.444 (−7.0%)▼	.613 (−2.6%)▼
−Rerank	.528 (−1.7%)▼	.462 (−3.0%)▼	.630 (−0.0%)◦

(b) Overall performance

The filtering and reranking modules in Aranea capture and exploit regularities in answers, which encode a type of "knowledge." The experiments reported here reveal that certain filters do have a significant impact on question answering performance, suggesting that data redundancy alone without a built-in bias for certain n-grams is insufficient to accurately answer factoid questions.

8. THE IMPACT OF OTHER SETTINGS

To complete our exploration of redundancy-based techniques, we describe additional experiments that examined a few more system settings. Although these runs did not directly relate to the two central theses of this work per se, they nevertheless addressed the broader question of "what really matters" in redundancy-based factoid question answering.

8.1 Answer Candidate Generation

Candidate answers to factoid questions were generated by exhaustively enumerating all n-grams from Web text snippets. Because no external resources were used, this knowledge-poor technique assumed that all n-grams were potential answers.

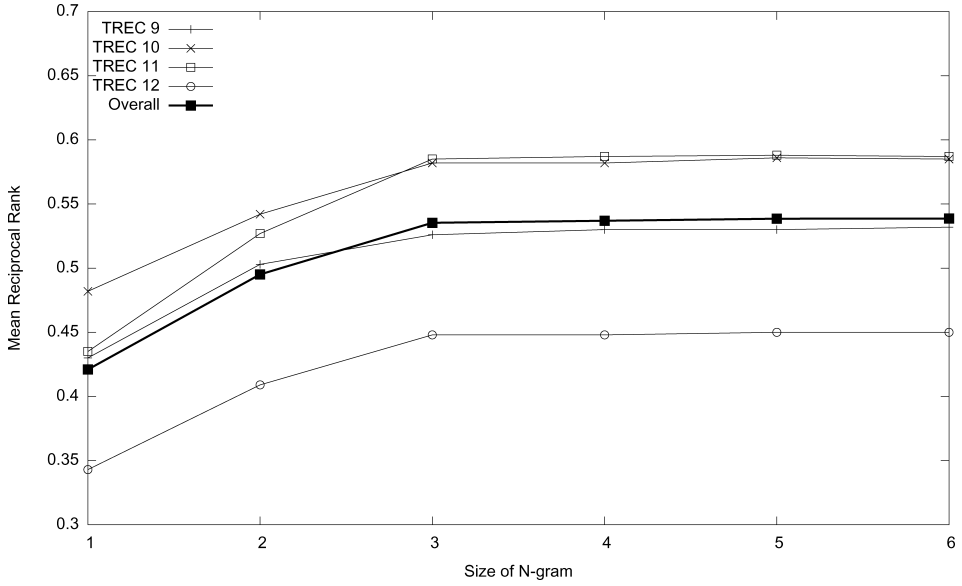

Fig. 12. MRR versus size of n-grams (z).

What was the effect of n-gram length on answer accuracy? Figure 12 shows mean reciprocal rank as a function of n-gram size (z) for all four test sets. These experiments employed all query types and snippets from both Google and Teoma. In the default configuration of Aranea, the n-gram size was set to 4 ($z = 4$), which means that all unigrams, bigrams, trigrams, and tetragrams were considered. The results showed that smaller values of z had an effect on mean reciprocal rank, but larger values ($z \geq 3$) did not. More formally, the Wilcoxon signed-rank tests reveals that differences in MRR between $z = 4$ and $z = 1, 2$ were statistically significant at the 1% level; no other differences were statistically significant.

Aranea's n-gram generation mechanism places inherent limitations on the types of answers that are generated. In particular, the size of the n-gram bounds the length of the candidates. As a result, it is impossible for Aranea to correctly answer questions such as "What does NAFTA stand for?" with its default configuration of $z = 4$. However, the experimental results shown in Figure 12 suggest that allowing for longer n-grams doesn't improve performance anyway—for such long answers, it is unlikely that the complete answer string will appear within a single text snippet. Hence, Aranea is likely to return a partial fragment, for example, "North American Free Trade." To cope with this issue, the AskMSR system implemented a "tiling mechanism" that created longer answers from overlapping fragments [Brill et al. 2001]. For example, the system would assemble "North American Free Trade Agreement" from "North American Free Trade" and "Free Trade Agreement." Although this technique can be effective at answering some questions, overgeneration is a problem, particularly for answers that are personal names. Because surnames can be written

either before or after the given name, tiling can generate inexact answers such as "Ray James Earl Ray."

On the other hand, it is not surprising that unigrams alone ($z = 1$) yield relatively good performance. Many factoid questions can be answered by a single term, for example, questions that ask for numbers such as "How many hexagons are on a soccer ball?" or "What year did Ellis Island open its doors to immigrants?" In addition, many acceptable "who" and "where" answers consist of single tokens, for example, last names and country names, respectively.

8.2 Answer Combination and Scoring

Since they are frequency-based, the n-gram generation and voting mechanisms tend to favor shorter strings. To counteract this tendency, Aranea applies a combining heuristic—each candidate answer receives a score boost equal to the sum of the scores of its component unigrams (see Figure 2 and Section 4.5). For example, the scores of the unigrams "James," "Earl," and "Ray" would be added to the score of the trigram "James Earl Ray," the correct answer to the question "Who shot Martin Luther King Jr.?" Although this combining heuristic has the danger of promoting candidate answers that have extraneous terms, for example, "James Earl Ray Dies," Section 4.5 discusses a number of different mechanisms that prevent this from happening in practice.

However, from an end-to-end applications point of view, we do not view answers with extraneous tokens as a serious concern because a previous study showed that users prefer answers embedded in some sort of context [Lin et al. 2003]. To the question "What Spanish explorer discovered the Mississippi River?" a response of "Hernando de Soto, a Spanish explorer, discovered the Mississippi River in 1541" is preferable to the exact answer "Hernando de Soto." Users are more confident in an answer's correctness when it is embedded in some context. Furthermore, surrounding text often provides answers to additional related questions, for example, "When did Hernando de Soto discover the Mississippi river?"

Furthermore, answer exactness is not a characteristic captured by current automatic evaluation resources. The answer patterns used in our experiments only indicate the presence or absence of correct answer strings, not their exact spans. Although it would be trivial to check if the pattern exactly spanned the system response, this would make the evaluation too restrictive, as the answer patterns were not designed to be used in this fashion. Therefore, it is difficult to objectively determine how often answers contain extraneous words, but visual examination of Aranea's output indicates that most answers are indeed exact.

To quantitatively assess the impact of the combining heuristic, we performed an ablation study with the component removed. Table XIV shows these results. The default configuration employed all reformulations and snippets from both Google and Teoma; the row marked − Combine indicates results without the combining module. Across all four test sets, removing this heuristic results in a drop of nearly 9% in MRR and 11% in C@1, both statistically significant at the 1% level. It appears that this heuristic is capturing a generalization about the distributional frequency of answers to factoid questions on the Web.

Table XIV. Impact of Various Answer Extraction Modules on Question Answering Accuracy

	TREC-9	TREC-10	TREC-11	TREC-12
MRR				
Default	.530	.582	.587	.448
−Combine	.482 (−9.1%)$^{\blacktriangledown}$	.546 (−6.2%)$^{\blacktriangledown}$	.523 (−10.9%)$^{\blacktriangledown}$	.414 (−7.6%)$^{\blacktriangledown}$
−*idf*	.522 (−1.5%)$^{\circ}$	.569 (−2.2%)$^{\circ}$	.565 (−3.7%)$^{\blacktriangledown}$	.445 (−0.7%)$^{\circ}$
C@1				
Default	.468	.517	.529	.392
−Combine	.414 (−11.7%)$^{\blacktriangledown}$	.480 (−7.2%)$^{\circ}$	.458 (−13.3%)$^{\blacktriangledown}$	.352 (−10.0%)$^{\triangledown}$
−*idf*	.461 (−1.5%)$^{\circ}$	.505 (−2.4%)$^{\circ}$	.500 (−5.4%)$^{\triangledown}$	.394 (+0.7%)$^{\circ}$
C@5				
Default	.627	.676	.678	.535
−Combine	.586 (−6.5%)$^{\blacktriangledown}$	.648 (−4.1%)$^{\circ}$	.626 (−7.8%)$^{\blacktriangledown}$	.512 (−4.4%)$^{\circ}$
−*idf*	.632 (+0.7%)$^{\circ}$	.670 (−0.9%)$^{\circ}$	.654 (−3.6%)$^{\triangledown}$	.522 (−2.4%)$^{\circ}$

(a) Individual test sets

	MRR	C@1	C@5
Default	.537	.477	.630
−Combine	.490 (−8.7%)$^{\blacktriangledown}$	.425 (−10.9%)$^{\blacktriangledown}$	.592 (−6.0%)$^{\blacktriangledown}$
−*idf*	.525 (−2.2%)$^{\blacktriangledown}$	.465 (−2.5%)$^{\circ}$	.620 (−1.6%)$^{\triangledown}$

(b) Overall performance

The prior distribution of terms is taken into account in the answer generation process by *idf* weighting of the candidate answers. In information retrieval, inverse document frequency has proven to be a simple yet effective method for capturing term importance—the measure encodes the insight that terms occurring in many documents should be given lower weight. Aranea scores each candidate by averaging the *idf* of its component unigrams. The effect of this weighting scheme is also shown in Table XIV. The default configuration employed all reformulations and snippets from both Google and Teoma; the row marked −*idf* shows results without the *idf* scoring module.

Experiments reveal that the impact of Aranea's term weighting method is rather small. No statistically significant differences are observed for C@1, although MRR and C@5 do show a small but significant overall drop without *idf* weighting.

Unfortunately, the generalizations that can be drawn from this result are rather limited. We have only shown the effect of one particular scoring technique, and have not made any effort to more exhaustively explore the large space of different weighting functions. More formal language modeling techniques are potentially applicable here as well—one could build an n-gram model of Web text and score answer candidates based on the probabilities of observing different answer strings, for example. Given the focus of this work, however, we did not conduct any experiments beyond the ones described here.

Based on the relatively minor impact of *idf* weighting, it is possible to speculate about the effects of candidate scoring methods in general. The primary goal of different scoring techniques is to differentiate candidates containing content terms from candidates containing non-content terms. This is necessary because the n-gram generation process does not identify well-formed

linguistic constituents, and therefore produces many implausible answers. However, Aranea employs other methods such as filtering to discard obviously problematic answer candidates. Therefore, the discriminative effect provided by different scoring algorithms may have less of an impact on question answering accuracy.

9. DISCUSSION

This work describes many experiments that examine the principles underlying redundancy-based factoid question answering, with the goal of determining "what really matters." In this section, we attempt to weave together the various threads discussed in the previous pages: first, a philosophical note; next, a discussion of the generalizations gleaned; and finally, a few words on more complex types of information needs.

9.1 Rationalism Versus Empiricism

The contrast between ontology-driven and redundancy-based question answering techniques reflects a divide between rational and empirical approaches to language technology applications. This debate is merely one episode in a much broader philosophical discourse dating back millennia; see, for example, Cahn et al. [1996]. Rationalists subscribe to the belief that certain types of knowledge are accessible via intuition alone, while others are knowable through deduction—the combination yields knowledge a priori, which is to say knowledge gained independently from sensory experience. Empiricists, on the other hand, claim that all knowledge is a posteriori (insofar as anything is "knowable"), or derived only from sensory experience.

Throughout the 1960s, 1970s, and into the 1980s, research in natural language processing was dominated by the rationalist tradition, and can be characterized by knowledge engineering, elaborate semantic representations, and carefully hand-crafted grammars. The end of the 1980s and the beginning of the 1990s saw a resurgence in empiricism, fueled by the availability of data, the explosive growth in computing power, and a growing emphasis on evaluation [Church and Mercer 1993; Brill and Mooney 1997]. Over the last decade, data-driven methods have become the dominant paradigm.

At first glance, the redundancy-based approach to factoid question answering appears to represent the logical extreme of empiricism. Data redundancy takes advantage of a prominent characteristic of the Web to overcome many troublesome issues in natural language processing (e.g., anaphora, inference, paraphrase, etc.). As previously discussed, this approach can be characterized as the philosophy of "data is all that matters," as espoused by Banko and Brill [2001]. In the limit, even specific learning algorithms are irrelevant—simply counting instances of observations should suffice.

Despite this starting point, our exploration of redundancy-based methods reveals a more complex picture. Throwing more data at the problem doesn't always help—in fact, retrieving too many snippets from the Web simply reinforces bad answers. Furthermore, many redundancy-based techniques implicitly encode heuristic knowledge about questions and candidate answers. For

example, reformulation rules encode linguistic knowledge about the structure of *wh*-questions. Filtering heuristics in the answer extraction phase of the processing pipeline encode knowledge about answer types. Removing these components leads to significant drops in answer accuracy, suggesting that knowledge still plays an important role in the redundancy-based approach.

Traditional and redundancy-based systems represent different tradeoff points in the broader space of knowledge-driven (rational) vs. data-driven (empirical) methods. With the aid of data redundancy, systems are able to achieve respectable accuracy with comparatively small amounts of knowledge engineering (recall that AskMSR was developed in less than 2 months). Ontology-driven methods, on the other hand, require large-scale knowledge engineering efforts and may suffer from some of the same problems that plagued early NLP systems built on hand-crafted grammars. Nonobvious dependencies between different system components cannot be easily anticipated and understood, and as a result small changes might lead to large, unexpected consequences. Most ontology-driven QA systems are large, monolithic structures that are difficult to describe and to build. For example, the most successful QA techniques reported in the TREC literature have yet to be successfully replicated by other groups. In contrast, while redundancy-based techniques do not achieve the same level of answer accuracy, the general approach has been widely adopted and its effectiveness has been independently verified. It appears that a system like Aranea represents a "sweet spot" in the tradeoff space of development effort versus performance.

Taking a higher-level view, we see that the solution space for factoid question answering is multidimensional and continuous. The traditional and redundancy-based approaches are merely convenient descriptors for points in this continuum—they are not mutually exclusive and there is no reason why systems cannot draw elements from both. For example, ontology-driven systems can employ voting techniques when there are multiple competing answer candidates (as many do). Similarly, redundancy-based systems can certainly benefit from formal ontological resources, although at the cost of added complexity. Furthermore, it is worth pointing out that there are alternative data-driven approaches to factoid question answering. Many researchers have conceptualized the task as a supervised machine learning problem and leveraged question–answer pairs (from the TREC data sets, from FAQ's mined from the Web, etc.) to learn methods for mapping questions to their answers, (e.g., Ittycheriah et al. [2000]; Berger et al. [2000]; Mann [2002]; Echihabi and Marcu [2003]; Agichtein et al. [2004]). An exploration of hybrid approaches that draw elements from all of these strategies is an interesting area for future work.

9.2 Generalizations

Although this work describes experiments with Aranea, our broader goal is to explore the general principles underlying redundancy-based factoid question answering. Although our system exemplifies the redundancy-based approach, it is important to uncover systematic generalizations, differentiating those from idiosyncrasies of specific system implementations. Distilling these higher-level

"take-away messages" has proven to be challenging, but we have been able to uncover some valuable lessons. To situate this discussion, we briefly reiterate the two main theses of this work:

—Stable characteristics of data redundancy allow factoid question answering systems to be built on external "black box" components.
—Despite embodying a data-driven approach, redundancy-based methods informally encode a substantial amount of knowledge in various heuristics.

Experiments described in Section 6, which study the performance characteristics of Web search engines under different conditions, speak primarily to the first issue. The fact that the same trends were observed across a variety of parameter settings suggests that our findings are valid generalizations about data redundancy. Overall, we discovered that both Google and Teoma behave in qualitatively similar ways across a number of confounding variables, exhibiting a performance curve that peaks rapidly, and then descends in a long tail. These results extend the work of Dumais et al. [2002] and reveal many interesting new findings.

Because data redundancy appears to be an inherent property of the Web, the exact choice of search engines and search engine parameters is less important than what one might think. This finding allows us to build systems that rely on external "black box" components. Thus, redundancy-based systems can take advantage of the retrieval infrastructure provided by existing commercial Web search engines.

Although earlier work touted data redundancy as the primary driver of performance, experiments with Aranea illustrate the important role that "knowledge" plays in the question answering process. This thesis was primarily explored in Section 7, which describes ablation experiments with Aranea's question analysis and answer extraction components.

In our system, question analysis is operationalized in terms of reformulation rules that convert questions into declarative statements that directly match anticipated answer forms (see Section 2.2). Due to the rather limited syntactic forms that factoid questions can take, a few pattern matching rules based on part-of-speech tags suffice to capture significant linguistic generalizations. These reformulations yield statistically significant increases in accuracy, although some of these gains can also be achieved by simply mining more diverse snippets from other search engines. We believe that this is a significant finding, as previous papers [Brill et al. 2001; Lin and Katz 2003] have overemphasized the role of direct pattern matching for answer extraction.

In general, the role of answer-type knowledge in redundancy-based factoid question answering can be intuitively understood in terms of a noisy channel model: given a question, the task is to reconstruct the answer (which has been "corrupted" by the channel). The various answer extraction heuristics encoded by Aranea in essence establish a prior distribution over all possible answers—the equivalent of the language model in the noisy-channel view. Similarly, the n-gram generation and voting processes establish the analog of the channel model. Our experiments demonstrate that having a good channel model isn't

enough: good answer accuracy can only be achieved by taking into account the a priori likelihood of answers.

Ultimately, what matters for redundancy-based factoid question answering? In short, we believe the answer is *diversity*. Redundancy is a powerful property, but having too much of similar text simply reinforces answers that have already been extracted (either correctly or incorrectly). In all of our experiments, giving Aranea "more of the same" did not improve performance, and in some cases, actually hurts. However, whenever the system was presented with qualitatively different text snippets (e.g., from different search engines, from different query types, etc.), significant improvements in answer accuracy were often observed. Similarly, settings that qualitatively affected the working set of candidate answers were important (e.g., different length *n*-grams, filters, etc.), while other settings that merely shuffled around candidates were less important (e.g., weight of exact reformulations, *idf-scoring*, etc.). A more fitting slogan for the redundancy-based approach is perhaps "the more diverse the data, the better." This conclusion makes intuitive sense, since data redundancy not only depends on the answer being stated multiple times, in multiple documents, but also *in multiple ways*.

9.3 Moving Beyond Factoids

Over the past few years, interest in question answering has shifted away from factoid questions to more complex information needs that cannot be addressed by short phrases. Consider the following examples:

—Who is Aaron Copland?
—How have South American drug cartels been using banks in Liechtenstein to launder money?
—What was the Pentagon panel's position with respect to the dispute over the U.S. Navy training range on the island of Vieques?

The first is an example of a so-called "definition" question, where the goal is to generate a profile of a person, entity, or event that integrates information from multiple sources. The second is an example of a "relationship" question, focused on ties (financial, familial, etc.) between different entities [Dang et al. 2006]. The last is an example of an "opinion" question, which might involve sentiment detection and analysis of language use. The growing interest in complex information needs is echoed by the development of query-focused multidocument summarization [Amigó et al. 2004] and formal evaluations of the task in recent Document Understanding Conferences [Dang 2005].

What is the role of factoid question answering given these developments? We believe that factoid systems will play an integral role within larger systems designed to handle complex questions. Consider answers to "Who is Aaron Copland?":

—American composer;
—wrote ballets and symphonies;
—born in Brooklyn, New York, in 1900;

—son of a Jewish immigrant;
—American communist;
—civil rights advocate.

For evaluating such complex questions, NIST adopts a methodology based on nuggets (i.e., "facts") that should be present in a good answer [Voorhees 2003]. As an example, the above "answer key" lists relevant nuggets for the question about Aaron Copland. From this, we can see that factoid question answering remains an important step in synthesizing an answer. In fact, definition questions can be viewed as simultaneously asking a whole series of factoid questions about the same entity (e.g., "When was he born?"; "What was his occupation?"; "Where did he live?" etc.), except that these questions are not known in advance [Prager et al. 2004]. Similarly, relationship and opinion questions can be decomposed into a series of smaller information needs, many of which might translate into factoid questions.

The understanding that complex information needs can be decomposed into a series of simpler questions is explicitly captured in the current TREC QA task definition. Since 2004, the main QA task at TREC has consisted of question series organized around topics (called "targets")—which can be people, organizations, events, or entities [Voorhees 2004]; cf. Kato et al. [2004]. Questions in a series inquire about different facets of a target, but are themselves either factoid or list questions. In addition, each series contains an explicit "other" question (always the last one), which can be paraphrased as "Tell me other interesting things about this target that I don't know enough to ask directly." These "other" questions represent that latest incarnation of definition questions.

As the state of the art advances, factoid question answering will be viewed less as an end-to-end application and more as a component within larger information systems. Extraction of individual facts can serve as the basis for automatically constructing "profiles" of entities, as in the case of definition questions. Individual facts can also serve as input to reasoning systems that infer answers not otherwise stated directly: a simple example of this is inferring a person's life span from birth and death dates. As the technology matures, we anticipate that factoid systems will become just another pluggable component within larger systems, much in the same manner that off-the-shelf document retrieval engines are currently used for a variety of applications. We hope that the open source release of Aranea will facilitate this process.

10. CONCLUSION

The redundancy-based approach to factoid question answering differs from traditional methods organized primarily around document retrieval and named-entity recognition technology, which require the support of large ontological resources. Although the label is convenient, the "redundancy-based approach" in actuality refers to a collection of different techniques that all aim to leverage the massive amounts of data available on the World Wide Web. Previously, many of these individual techniques have only been evaluated in end-to-end systems; as a result, the contributions of different components and the impact of parameter settings are not well known. The primary contribution of this

work is a detailed, systematic exploration of the principles and assumptions underlying redundancy-based techniques, supported by evidence from ablation and contrastive experiments. Ultimately, we hope that our findings provide a deeper understanding of data redundancy and guidance for future developers of factoid question answering systems.

ACKNOWLEDGMENTS

I am grateful to Sue Dumais and three anonymous referees for their challenging questions, detailed comments, and thorough reviews—they have helped make this article significantly better. I would also like to thank Ken Church for engaging philosophical discussions. All other errors and failings in this article are, of course, my own.

REFERENCES

AGICHTEIN, E. AND GRAVANO, L. 2000. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the 5th ACM International Conference on Digital Libraries* (DL 2000). 85–94.

AGICHTEIN, E., LAWRENCE, S., AND GRAVANO, L. 2004. Learning to find answers to questions on the Web. *ACM Trans. Int. Tech. 4*, 2, 129–162.

AMIGÓ, E., GONZALO, J., PEINADO, V., PEÑAS, A., AND VERDEJO, F. 2004. An empirical study of information synthesis task. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics* (ACL 2004). 207–214.

AZARI, D., HORVITZ, E., DUMAIS, S., AND BRILL, E. 2004. Actions, answers, and uncertainty: A decision-making perspective on Web-based question answering. *Inform. Process. Manage. 40*, 5, 849–868.

BANKO, M. AND BRILL, E. 2001. Scaling to very very large corpora for natural language disambiguation. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics* (ACL 2001). 26–33.

BAR-ILAN, J. 2002. Methods for measuring search engine performance over time. *J. Amer. Soc. Inform. Sci. Techn. 53*, 4, 308–319.

BERGER, A., CARUANA, R., COHN, D., FREITAG, D., AND MITTAL, V. 2000. Bridging the lexical chasm: Statistical approaches to answering finding. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (SIGIR 2000). 192–199.

BRILL, E. 1995. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computat. Ling. 21*, 4, 543–565.

BRILL, E., DUMAIS, S., AND BANKO, M. 2002. An analysis of the AskMSR question-answering system. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing* (EMNLP 2002). 257–264.

BRILL, E., LIN, J., BANKO, M., DUMAIS, S., AND NG, A. 2001. Data-intensive question answering. In *Proceedings of the Tenth Text REtrieval Conference* (TREC 2001). 393–400.

BRILL, E. AND MOONEY, R. J. 1997. An overview of empirical natural language processing. *AI Mag. 18*, 4, 13–24.

CAFARELLA, M. J., DOWNEY, D., SODERLAND, S., AND ETZIONI, O. 2005. KnowItAll: Fast, scalable information extraction from the Web. In *Proceedings of the 2005 Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing* (HLT/EMNLP 2005). 563–570.

CAHN, S. M., KITCHER, P., SHER, G., AND MARKIE, P. J. 1996. *Reason at Work: Introductory Readings in Philosophy*, 3rd ed. Hardcourt Brace College Publishers, Fort Worth, TX.

CHU-CARROLL, J., CZUBA, K., PRAGER, J., AND ITTYCHERIAH, A. 2003. In question answering, two heads are better than one. In *Proceedings of the 2003 Human Language Technology Conference and the North American Chapter of the Association for Computational Linguistics Annual Meeting* (HLT/NAACL 2003). 24–31.

CHURCH, K. W. AND MERCER, R. L. 1993. Introduction to the special issue on computational linguistics using large corpora. *Computat. Lingu. 19*, 1, 1–24.

CLARKE, C., CORMACK, G., AND LYNAM, T. 2001a. Exploiting redundancy in question answering. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (SIGIR 2001). 375–383.

CLARKE, C., CORMACK, G., LYNAM, T., LI, C., AND MCLEARN, G. 2001b. Web reinforced question answering (MultiText experiments for TREC 2001). In *Proceedings of the Tenth Text REtrieval Conference* (TREC 2001). 673–679.

CUI, H., SUN, R., LI, K., KAN, M.-Y., AND CHUA, T.-S. 2005. Question answering passage retrieval using dependency relations. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development of Information Retrieval* (SIGIR 2005). 400–407.

DANG, H. 2005. Overview of DUC 2005. In *Proceedings of the 2005 Document Understanding Conference* (DUC 2005) at NLT/EMNLP 2005.

DANG, H., LIN, J., AND KELLY, D. 2006. Overview of the TREC 2006 question answering track. In *Proceedings of the Fifteenth Text REtrieval Conference* (TREC 2006).

DUMAIS, S., BANKO, M., BRILL, E., LIN, J., AND NG, A. 2002. Web question answering: Is more always better? In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (SIGIR 2002). 291–298.

ECHIHABI, A. AND MARCU, D. 2003. A noisy-channel approach to question answering. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics* (ACL 2003). 16–23.

FLEISCHMAN, M., HOVY, E., AND ECHIHABI, A. 2003. Offline strategies for online question answering: Answering questions before they are asked. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics* (ACL 2003). 1–7.

FUKUMOTO, J., KATO, T., AND MASUI, F. 2002. Question Answering Challenge (QAC-1): An evaluation of question answering task at NTCIR Workshop 3. In *Proceedings of the Third NTCIR Workshop on Research in Information Retrieval, Automatic Text Summarization and Question Answering*.

HARABAGIU, S., MOLDOVAN, D., PAŞCA, M., MIHALCEA, R., SURDEANU, M., BUNESCU, R., GÎRJU, R., RUS, V., AND MORĂRESCU, P. 2000a. FALCON: Boosting knowledge for answer engines. In *Proceedings of the Ninth Text REtrieval Conference* (TREC-9). 497–506.

HARABAGIU, S., PAŞCA, M., AND MAIORANO, S. 2000b. Experiments with open-domain textual question answering. In *Proceedings of the 18th International Conference on Computational Linguistics* (COLING 2000). 292–298.

HILDEBRANDT, W., KATZ, B., AND LIN, J. 2004. Answering definition questions with multiple knowledge sources. In *Proceedings of the 2004 Human Language Technology Conference and the North American Chapter of the Association for Computational Linguistics Annual Meeting* (HLT/NAACL 2004). 49–56.

HIRSCHMAN, L. AND GAIZAUSKAS, R. 2001. Natural language question answering: The view from here. *Nat. Lang. Eng. 7*, 4, 275–300.

HOVY, E., GERBER, L., HERMJAKOB, U., JUNK, M., AND LIN, C.-Y. 2000. Question answering in Webclopedia. In *Proceedings of the Ninth Text REtrieval Conference* (TREC-9). 655–664.

ITTYCHERIAH, A., FRANZ, M., ZHU, W.-J., AND RATNAPARKHI, A. 2000. IBM's statistical question answering system. In *Proceedings of the Ninth Text REtrieval Conference* (TREC-9). 258–264.

KATO, T., FUKUMOTO, J., MASUI, F., AND KANDO, N. 2004. Handling information access dialogue through QA technologies—a novel challenge for open-domain question answering. In *Proceedings of the HLT-NAACL 2004 Workshop on Pragmatics of Question Answering*. 70–77.

KATZ, B. 1997. Annotating the World Wide Web using natural language. In *Proceedings of the 5th RIAO Conference on Computer Assisted Information Searching on the Internet* (RIAO 1997). 136–155.

KATZ, B., FELSHIN, S., YURET, D., IBRAHIM, A., LIN, J., MARTON, G., MCFARLAND, A. J., AND TEMELKURAN, B. 2002. Omnibase: Uniform access to heterogeneous data for question answering. In *Proceedings of the 7th International Workshop on Applications of Natural Language to Information Systems* (NLDB 2002). 230–234.

KWOK, C., ETZIONI, O., AND WELD, D. S. 2001. Scaling question answering to the Web. *ACM Trans. Inform. Syst. 19*, 3, 242–262.

LIGHT, M., MANN, G. S., RILOFF, E., AND BRECK, E. 2001. Analyses for elucidating current question answering technology. *Nat. Lang. Eng. 7*, 4, 325–342.

LIN, J. 2005. Evaluation of resources for question answering evaluation. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (SIGIR 2005). 392–399.

LIN, J. AND DEMNER-FUSHMAN, D. 2005. Automatically evaluating answers to definition questions. In *Proceedings of the 2005 Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing* (HLT/EMNLP 2005). 931–938.

LIN, J., FERNANDES, A., KATZ, B., MARTON, G., AND TELLEX, S. 2002. Extracting answers from the Web using knowledge annotation and knowledge mining techniques. In *Proceedings of the Eleventh Text REtrieval Conference* (TREC 2002).

LIN, J. AND KATZ, B. 2003. Question answering from the Web using knowledge annotation and knowledge mining techniques. In *Proceedings of the Twelfth International Conference on Information and Knowledge Management* (CIKM 2003). 116–123.

LIN, J. AND KATZ, B. 2006. Building a reusable test collection for question answering. *J. Amer. Soc. Inform. Sci. Tech. 57*, 7, 851–861.

LIN, J., QUAN, D., SINHA, V., BAKSHI, K., HUYNH, D., KATZ, B., AND KARGER, D. R. 2003. What makes a good answer? The role of context in question answering. In *Proceedings of the Ninth IFIP TC13 International Conference on Human-Computer Interaction* (INTERACT 2003). 25–32.

LOWE, J. B. 2000. What's in store for question answering? (Invited talk.) In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora* (EMNLP/VLC-2000).

MAGNINI, B., ROMAGNOLI, S., VALLIN, A., HERRERA, J., PEÑAS, A., PEINADO, V., VERDEJO, F., AND DE RIJKE, M. 2004. The multiple language question answering track at CLEF 2003. In *Comparative Evaluation of Multilingual Information Access Systems: 4th Workshop of the Cross-Language Evaluation Forum, CLEF 2003, Trondheim, Norway, August 21–22, 2003, Revised Selected Papers*, C. Peters, J. Gonzalo, M. Braschler, and M. Kluck, Eds. Lecture Notes in Computer Science, vol. 3237. Springer, Berlin, Germany, 471–486.

MANN, G. 2002. Learning how to answer question using trivia games. In *Proceedings of the 19th International Conference on Computational Linguistics* (COLING 2002).

MITTENDORF, E. AND SCHÄUBLE, P. 1994. Document and passage retrieval based on Hidden Markov Models. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (SIGIR 1994). 318–327.

MOFFAT, A., SACKS-DAVIS, R., WILKINSON, R., AND ZOBEL, J. 1993. Retrieval of partial documents. In *Proceedings of the Second Text REtrieval Conference* (TREC-2). 181–190.

MOLDOVAN, D., PAŞCA, M., HARABAGIU, S., AND SURDEANU, M. 2002. Performance issues and error analysis in an open-domain question answering system. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics* (ACL 2002). 33–40.

NTOULAS, A., CHO, J., AND OLSTON, C. 2004. What's new on the Web? The evolution of the Web from a search engine perspective. In *Proceedings of the 13th International World Wide Web* Conference (WWW 2004). 1–12.

PRAGER, J., BROWN, E., AND CODEN, A. 2000. Question-answering by predictive annotation. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (SIGIR 2000). 184–191.

PRAGER, J., CHU-CARROLL, J., AND CZUBA, K. 2004. Question answering using constraint satisfaction: QA–by–Dossier–with–Constraints. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics* (ACL 2004). 574–581.

RAVICHANDRAN, D. AND HOVY, E. 2002. Learning surface text patterns for a question answering system. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics* (ACL 2002). 41–47.

ROBERTSON, S. 1977. The probability ranking principle in IR. *J. Documentat. 33*, 4, 294–304.

ROBERTSON, S. 2004. Understanding inverse document frequency: On theoretical arguments for IDF. *J. Documentat. 60*, 5, 503–520.

SALTON, G., ALLAN, J., AND BUCKLEY, C. 1993. Approaches to passage retrieval in full text information systems. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (SIGIR 1993). 49–58.

SRIHARI, R. AND LI, W. 1999. Information extraction supported question answering. In *Proceedings of the Eighth Text REtrieval Conference* (TREC-8). 185–196.

TELLEX, S., KATZ, B., LIN, J., MARTON, G., AND FERNANDES, A. 2003. Quantitative evaluation of passage retrieval algorithms for question answering. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (SIGIR 2003). 41–47.

VOORHEES, E. 2001. Overview of the TREC 2001 question answering track. In *Proceedings of the Tenth Text REtrieval Conference* (TREC 2001). 42–51.

VOORHEES, E. 2002. Overview of the TREC 2002 question answering track. In *Proceedings of the Eleventh Text REtrieval Conference* (TREC 2002). 57–68.

VOORHEES, E. 2003. Overview of the TREC 2003 question answering track. In *Proceedings of the Twelfth Text REtrieval Conference* (TREC 2003). 54–68.

VOORHEES, E. 2004. Overview of the TREC 2004 question answering track. In *Proceedings of the Thirteenth Text REtrieval Conference* (TREC 2004). 52–69.

VOORHEES, E. AND TICE, D. 1999. The TREC-8 question answering track evaluation. In *Proceedings of the Eighth Text REtrieval Conference* (TREC-8). 83–106.

VOORHEES, E. AND TICE, D. 2000. Building a question answering test collection. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (SIGIR 2000). 200–207.

ZOBEL, J., MOFFAT, A., AND SACKS-DAVIS, R. 1995. Efficient retrieval of partial documents. *Inform. Process. Manage. 31*, 3, 361–377.

Received November 2005; revised June 2006; accepted October 2006