Applying Question Answering Technology to Locating Malevolent Online Content

Dmitri Roussinov[a,*], José A. Robles-Flores[a,b]

* Corresponding author. E-mail address: Dmitri.Roussinov@asu.edu

[a] Department of Information Systems, W.P. Carey School of Business, Arizona State University, USA

[b] ESAN University, Lima, Peru

Abstract

We have empirically compared two classes of technologies capable of locating potentially malevolent online content: 1) popular keyword searching, currently widely used by law enforcement and general public, and 2) emerging question answering (QA). The Google search engine exemplified the first approach. To exemplify the second, we further advanced the pattern based probabilistic QA approach and implemented a proof-of-concept prototype that was capable of finding web pages that provide the answers to the given questions, including non factual ones (e.g. "How to build a pipe bomb?"). The answers to those question typically indicate the presense of malevolent content. Our findings suggest that QA technology can be a good addition to the traditional keyword searching for the task of locating malevolent online content and, possibly, for a more general task of interactive online information exploration.

Keywords: Information systems security, information retrieval, question answering, world wide web.

Introduction

After the September 11 attacks, the world started to pay close attention to its vulnerable assets, one of which is undoubtedly the Internet -- the backbone of modern information superhighway. Making cyber infrastructure resilient to any attacks or misuse became a priority for scientists and funding agencies [17]. However, the media still reports numerous cases of government web sites "defaced" or shut down by hackers [32]. In addition, the proliferation of illegal spamming, computer viruses, identity theft, software piracy, and fraudulent schemes has threatened the trust behind electronic means of communication to the

degree of becoming a threat to the national cyber infrastructure [35]. A recent example reported in popular press [13] is the shadowcrew.com web site, which acted as a one-stop shop for false documents and information stolen by hacking into computers. The site also posted hacking methods and maintained a forum for hackers to exchange their ideas and techniques.

We define the content that can facilitate committing crimes (not necessarily in cyberspace) as *malevolent*. The shadowcrew.com example clearly illustrates that *malevolent content frequently co-exists with illicit activities*. Law enforcement and public watchdogs actively locate, monitor, and sometimes act on such content [38]. For example, the "sting" operation performed by the US Secret Service on shadowcrew.com resulted in 28 arrests across different countries [13].

In order to become subject to monitoring and legal actions, the malevolent content first needs to be located. However, while the most notorious sites can be easily found, many less known sites can remain undetected [22]. Since the resources of law enforcement and researchers are limited, it may take a long time before potentially malevolent content is discovered and becomes subject to monitoring or other actions. One practical way to search the content of the Internet, available for law enforcement and public watchdogs, is by using a search engine, such as Google, AOL, or MSN. The agents need to come up with and maintain a list of keywords that would potentially uncover the parts of "Dark Web", run several queries and laboriously analyze the search results [22].

The algorithms that search engines use are based on lexical match (so called "bag of words" approach) in which the pages and queries are represented as sets (bags) of words. The use of this approach results in the well-known problem of information overload on the web [16] [23] [33] considering that the web has more than 8 billion pages, the vast majority of them legitimate and harmless. Performing Google (or other search engine) search on the topic of "hacking" and "phishing" results in thousands of pages from the news media (e.g. cnn.com) or political discussion forums (e.g. soc.culture.usa) since the search engines' algorithms locate the content based on the lexical (word) match to the query and the popularity of the web sites [2], thus mostly overlooking "shady" portions of the web. It is the level of technical detail that can distinguish innocuous pages from harmful ones, news from "how-to" manuals. However,

this level of depth cannot be picked by lexical matching or link analysis since they have never been designed for that purpose.

An emerging alternative to keyword-based web searching is automated question answering (QA). A typical QA system would take the question in a natural language such as "How can I guess a password?" and return an answer such as "You can use a password dictionary to guess passwords" and/or a link to a source page that provides the answer. Recent breakthroughs in the QA technologies have been reported [37] and are briefly reviewed in the next section.

Several research groups have made publicly available their demonstration systems capable of finding the answers on the World Wide Web: Language Computer (www.languagecomputer.com), AnswerBus (www.answerbus.com), NSIR (http://tangra.si.umich.edu/clair/NSIR/html/nsir.cgi) and ASU QA Demo (http://qa.wpcarey.asu.edu). This has attracted attention from the media. For example, Information Week recently mentioned ASU QA system as one of the most promising directions in the "Search of Tomorrow." [7]

We believe that QA technology may be a good addition to the keyword searching for locating malevolent online content, because it seeks the pages that may provide answers to the questions entered by law enforcement agents or public watchdogs, e.g. "How to break into a network?", "How to steal a credit card number?", etc. This accurate pinpointing should be more effective than lexical match and popularity analysis currently used by commercial search engines.

In spite of recent breakthroughs in QA technology and the promise it carries for a number of applications, no experiments have been performed to compare Web QA tools and popular search engines in which human users apply the tools to accomplish search tasks interactively, although a number of studies compared QA tools with search engines tested in batch mode [20] [21]. This is true for the more general domain of information exploration, and to the more specialized domain of security applications. As a result, *it is not entirely clear if and by how much QA technology can help locating malevolent online content in addition to the currently available keyword searching.* This lack of empirical comparison and

the desire to provide law enforcement with more effective tools to locate and monitor malevolent content have motivated our study. We report the following contributions:

- We have empirically compared two classes of technologies capable of locating potentially malevolent online content: 1) popular keyword searching, currently widely used by law enforcement and the general public, and 2) emerging question answering (QA) technology. Our specially created QA prototype system exemplifies this approach, while Google search engine represents the first.
- We have adapted and further developed pattern based probabilistic QA technology for the non-factual types of questions such as "How to", "How do I", etc. since we also empirically discovered throughout our study that those types of questions are used very frequently in locating that type of content.

In the following section, we present a review of the related literature. The section on "Algorithms and Implementations" presents the prototype used in this research. The "Empirical Comparison" section identifies the research questions, the methodology used and the results obtained. Then, we present our "Conclusions, Limitations and Future Research Directions."

Literature Review

Malevolent Content on the Web

The review on the proliferation of the malevolent content online, such as potentially assisting terrorists or cyber criminals, can be found in the special report by the United States Institute of Peace [38]. The report includes a number of vivid examples, such as the one about David Copeland who planted nail bombs in three different areas of London, which killed 3 people and injured 139. At his trial, he revealed that he had learned his deadly techniques from the Internet, downloading *The Terrorist's Handbook* and *How to Make Bombs: Book Two.*

Another example mentioned a brilliant chemistry student, calling himself "RC", who discussed bomb-making techniques with other enthusiasts on a Finnish Internet website devoted to bombs and explosives. RC set off a bomb that killed seven people, including him, in a crowded shopping mall. The website

frequented by RC, known as the Home Chemistry Forum, was shut down by its sponsor, an undisclosed computer magazine. But a backup copy was immediately posted again on a read-only basis.

The Guardian [34] reported that copies of the "best-known bomb-making guide" called "The Anarchist Cookbook" [18], offering instructions on everything from making an atomic bomb to devising a rocket launcher "for under $5", are easily available online.

The other type of malevolent content widespread online is the one that facilitates committing cyber-crimes by providing instructions online to "would be" criminals, means for "hackers" to exchange their ideas and sell illegally acquired information such as credit card numbers, passwords, and personal profiles. The famous shadowcrew.com example mentioned in our introduction shows that illicit activities frequently co-exist with malevolent content.

For this reasons, malevolent content becomes subject to monitoring by law-enforcement, public watchdogs and researchers. For example, The Dark Web Portal project at the University of Arizona [6] [22], supported by the National Science Foundation (NSF), helps researchers locate, collect, and analyze the alternate side of the Web which is used by terrorist and extremist groups. The project studies various methods of locating the malevolent online content. The researchers reported that the activities to locate these dangerous contents need substantial manual work in identifying seed pages (starting points), finding the query terms and filtering irrelevant results. Thus, there is a growing need for more powerful tools to locate potentially malevolent content, which is the focus of our paper. Although we limited the scope of our study to the content that facilitates cyber crime, since the techniques that we studied are not domain specific we believe that our results can be generalized to virtually any type of malevolent content.

Automated Question Answering (QA)

The interaction in a natural language form has been a dream of artificial intelligence (AI) since the invention of computers and even foreseen earlier by creative imagination (e.g. Wells, "The Time Machine," [39]). Recent advances in Natural Language Processing (NLP) and AI in general have approached this dreamed world to the point where it mixes with reality. Several well-known futurists

believe that computers will reach capabilities comparable to human reasoning and understanding of languages by the year 2020 [14].

The National Institute of Standards and Technology (NIST) organizes the annual Text Retrieval and Evaluation Conference (TREC) in which researchers and commercial companies developing various human language technologies compete in such classical tasks as document retrieval, and newer tasks such as question answering, novelty and topic detection, interactive and Web searching [36]. This reflects the information access paradigm shifting from keyword based search to natural language driven navigation.

Systems participating in TREC have to identify exact answers to so called *factual* questions (or *factoids*), such as *who, when, where, what,* etc., list questions (*What companies manufacture rod hockey games?*) and definitions (*What is bulimia?*). In order to answer such questions, a typical TREC QA system would: (a) transform the user query into a form it can use to search for relevant documents (web pages), (b) identify the relevant passages within the retrieved documents that may provide the answer to the question, and (c) identify the most promising candidate answers from the relevant passages. Most of TREC QA systems are designed based on techniques from natural language processing (NLP), information retrieval (IR) and computational linguistics (CL). For example, Falcon [9], one of the most successful systems, is based on a pre-built hierarchy of dozens of semantic types of expected answers (*person, place, profession, date, etc.*), complete syntactic parsing of all potential answer sources, and automated theorem proving to validate the answers. Naturally, those modules are time consuming and difficult to integrate within other knowledge management or information awareness systems.

In contrast to the NLP-based approaches that rely on laboriously created linguistic resources, "shallow" approaches that use only simple pattern matching have been recently successfully tried, e.g. the system from InsightSoft [30] won the 1st place in 2002 and the 2nd place in 2001 TREC competitions. However, none of the best performing (non-web) systems is publicly available for evaluation or for inclusion in a research prototype.

Web Question Answering

Past studies [19] have indicated that the current WWW search engines, especially those with very large indexes like Google, offer a very promising source for open domain (not limited to any topics or sources) question answering. Also, an analysis of requests processed by the Excite search engine [19] found that 8.4% of the queries were in the form of questions and a more significant percentage with a certain question in mind. Once the capability of commercial search engines includes QA, the proportion will most likely increase.

This explains why there are numerous efforts are under way porting and adapting existing QA techniques to the much larger context of the World Wide Web. AskJeeves (www.ask.com), a public company positions itself as the pioneer of Web QA. However their knowledge sources are limited to a small set of specially created databases (e.g. geographical locations). When answers are not found there, AskJeeves reroutes the question as a simple keyword query to a general purpose search engine (Theoma, http://www.teoma.com/). In spite of this limitation, AskJeeves was recently purchased by IAC/InterActiveCorp for $1.9 billion [7], a price comparable even with the value of Google stock at the time.

A relatively complete, general-purpose, web-based QA system, called NSIR, was presented in Radev et al. [20] and Radev et al. [21]. Dumais et. al [8] presented another open-domain Web QA system that applies simple combinatorial permutations of words (so called "re-writes") to the snippets returned by Google and a set of 15 handcrafted semantic filters to achieve a striking accuracy: Mean Reciprocal Rank (MRR) of *0.507*, which can be roughly interpreted as "in average" the correct answer being the second answer found by the system.

Roussinov and Robles-Flores [24] expanded work by Dumais et al. [8] by automated identification and training of patterns, triangulation and using trainable semantic filters instead of manually created ones. The pattern based approach has additional advantages over deep NLP approaches for locating content on the Web because it can look for grammatically irregular sentences or combinations of headings followed

by answer paragraphs. For example, the system presented by Roussinov and Robles-Flores [24], once trained and given a question "How to hack into computer networks," would be looking for such patterns as "Re: How to hack into computer networks," "How To Hack Into Computer Networks Tutorial," "Introduction to hacking into computer networks," etc. Their system can be automatically trained on the set of questions and answers to build the set of patterns for each question type. A more detailed description follows in the next section.

Algorithms and Implementations

Overview

The pattern based QA approach used in our study has been described by Roussinov and Robles-Flores [24]. They extended the prior research by Dumais et. al. [8]. Use of their system offered the following important advantages:

1) Their system is entirely transparent in the sense that all their algorithms are detailed in their prior publications. It does not depend on manual tuning, is completely trainable from examples, thus the results can be easily reproduced in other studies.

2) It allows searching the entire Web, which is essential in our study, as opposed to using a system like AskJeeves that delivers answers only from a specially constructed database.

When searching for an answer to a question (e.g. *"Who is the CEO of IBM?"*) their approach looks for matches to certain patterns. For example *"The CEO of IBM is Samuel Palmisano."* matches the pattern *"\Q is \A",* where *\Q* is a question part (*"The CEO of IBM"*) and *\A = "Samuel Palmisano"* is the text that forms a candidate answer. The approach automatically creates and trains up to *200* patterns for each type of a question (e.g. *what is, what was, where is*, etc.) based on a training set of given question-answer pairs. Through training, each pattern is assigned the probability that the matching text contains the correct answer. This probability is used in ranking the candidate answers. *\A, \Q, \p* (punctuation mark), \s (beginning of a sentence) and * (wildcard that matches any words) are the only special symbols currently used in the pattern language.

Training of the system has been performed on the questions and answers for the Text Retrieval and Evaluation Conference (TREC) [36]. There are approximately 1,500 questions in the set so far, with the most popular question types like *what is, when was, where is* represented by several hundreds of examples. Since the questions are from open (non-restricted) domain and the training mechanism does not depend on any specific terminology, we expected the training results to be generalizable to the types of tasks involved in our study.

Steps Involved

Answering the question "Who is the CEO of IBM?" demonstrates the steps of our algorithm:

Type Identification. The question itself matches the pattern *who is \Q,* where *\Q = "the CEO of IBM"* is the question part.

Query modulation converts each answer pattern (e.g. *\A * became \Q*) into a query for a general purpose search engine (GPSE). For example, the Google query will be *+"became" +"the CEO of IBM"* . The plus sign (+) requests the entire phrase within the quotes to be present in the retrieved web pages. The query is sent to the GPSE and the top 100 retrieved snippets (or full text of pages if requested) are scanned for answer pattern matches.

Answer Matching. The sentence "Samuel Palmisano recently became the CEO of IBM." would result in a match and produce a candidate answer "Samuel Palmisano recently". When fewer than the specified number of matches are found, the system resorts to the "fall-back" mechanism by sending the question to the underlying search engine (GPSE) and forming candidate answers directly from the returned snippets.

Answer Detailing produces more candidate answers by forming sub-phrases from the initial candidate answers. The sub-phrases do not exceed 3 words (not counting "stop words" such as *a, the, in, on*) and do not cross punctuation marks. In our example, the detailed candidate answers would be *Samuel, Palmisano, recently, Samuel Palmisano, Palmisano recently*.

Triangulation. The candidate answers are triangulated (confirmed or disconfirmed) against each other as explained by the following intuitive example. Imagine that we have two candidate answers for the

question *"What was the purpose of Manhattan Project?"* 1) *"To develop a nuclear bomb"* and 2) *"To create a nuclear weapon."* Those two answers should reinforce (triangulate) each other since they are semantically similar. This is achieved by recognizing sharing the same word (*nuclear*) and the pair wise semantic similarity between the words *bomb*/*weapon* and *develop*/*create*. Similarly, in the above example "*who is the CEO of IBM*", the candidate answers *recently* and *Palmisano recently* receive less support and are unlikely to become the top candidates than the candidate answer *Samuel Palmisano*. A spurious candidate answer (e.g. *Bill Gates* from a sentence *Bill Gates did not become the CEO of IBM*) would not receive much support from other candidate answers during triangulation.

The final answer score is the result of the original score associated with each pattern and the triangulation process. The answers are presented to the user ordered by the final score and within the context of the sentence where they were found followed by the link to the source page.

Pattern Identification

Although in the original approach patterns were trained completely automatically [24] for factual questions, there are no existing training sets or training algorithms developed for non-factual ones. To train the QA tool for non-factual questions, we adopted a semi-automatic process. The collection of Frequently Asked Questions files available from the FAQ Finder Project [3] have been manually reviewed by the software engineers involved in the implementation and 40 patterns were manually identified. The entire process took approximately 1 man-hour. A few more patterns were added later during testing of the prototype. We believe it is entirely possible to create heuristic algorithms to automatically identify patterns for many types of non factual questions from the examples similar to FAQ collection, but we left that for future research. Table 1 lists the patterns used in our study. The patterns typically do not have the answer part (*A*) because, for a non-factual question, the entire matching sentence becomes a candidate answer. We also involved several patterns that are based on the headings followed by the answer text. The same patterns were used for "How do I \Q", and "How can I \Q," "How can someone \Q" and "How is \Q done" questions. For the other type of questions, including the factual

ones (e.g. "What is a password dictionary?"), our system was identical to the original one in Roussinov and Robles-Flores [24].

Insert Table 1

Scalability and Responsiveness

Since our objective was to compare the two approaches, we were not concerned with real-time responsiveness of our prototype. Our QA system, implemented specifically for this study, typically finds a pre-specified number of answers (40) within a minute. The important design decision was to parallelize the querying of the underlying search engine (Google in our case). Although it still remains the bottleneck, it can be optimized even further by using multiple workstations, as for example has been successfully demonstrated for non Web QA in Surdeanu et al. [31]. Another possible solution is to have direct access to the search engine's index and cache, which may be, for example, feasible when a QA system is an internal part of it.

Empirical Comparison

Research Questions and Methods

Our main objective was to compare two different approaches to locating malevolent web content (keyword search vs. pattern based question answering) exemplified by the following two tools: ASU QA Demo (also referred throughout our paper as *QA tool*) and Google. We have selected Google due to its leading position in the search engine market (e.g. according to www.searchwnginewatch.com) and simplicity of use. Other popular search engines such as Alta Vista, Yahoo, AOL, and MSN have very similar interfaces and query languages. They all essentially treat a user query as a "bag of words." Google also served as the underlying general purpose search engine (GPSE) for our QA tool.

The theoretical value of our comparison can be illustrated in terms of the 4-step information exploration model [29] which considers interactive information seeking consisting of the following steps: 1) Formulation: expressing the search task. 2) Action: launching the search through a query. 3) Review of results: reading messages and outcomes resulting from the search. 4) Refinement: revising the query if

needed and returning to step 2. From this perspective, our comparison tested if our pattern matching layer was positively influencing the effectiveness of step 1 by allowing a more natural formulation of the task through a question, and step 2 by improving the overall quality of the retrieved results due to the use of a different ranking mechanism (the likelihood of answering rather likelihood of being relevant). Thus, our first ("task level") research question was the following:

Q1: Does question answering (QA) technology enhance the ability to locate malevolent content if compared to traditional keyword search?

We know from the prior studies in the area of interactive information exploration [28] [15] [23], that the users' ability to locate the desired content is affected by the following major factors: 1) relevance of the results returned by the system 2) the responsiveness of the system (average time delay between input and output) and 3) the ability to find the content within the search results. The latter, in turn, depends on a) the presentation format, and b) the user skills.

To simplify our empirical challenge and avoid the impact of the third factor (ability to find within the search results), we enforced the same presentational format for the two compared approaches. We deliberately implemented our QA system in such a way that it presents results in the same way as the general purpose search engines do: showing snippets of the pages instead of the answer sentences (Figure 1). By doing so, we have artificially "handicapped" our QA system, but also have been able to leave investigating the effects of the presentation (snippets vs. answers) for future research.

Thus, if the responsiveness was not an issue (both tools are quick enough), we would expect that the improved relevance of the returned pages would imply the enhanced ability to find the desired content. We believed that the web pages found by QA technology (pattern matching) would return more relevant pages (more likely to contain the answers) than those retrieved by search engines. We posed our second ("direct") research question as follows:

Q2: Does using question answering (QA) technology improve the relevance of the retrieved pages?

To address two of the above research questions accordingly, we have performed our study in two phases: 1) *The Task Level Phase*, where the study volunteers came up with their own questions and attempted to

answer them using one of the tools, once for each question and 2) *The Blind Evaluation Phase*, where the volunteers only evaluated the results retrieved by each of the tools without knowing which one had retrieved them. The second phase excluded the factor of responsiveness (time), because its effect was not that interesting from a scientific standpoint: any desired responsiveness can be attained by just using more resources (e.g. larger number of parallel servers) as already discussed in the "Algorithms and Implementations" section. Also, because the design of the second phase ensured that each question was tested by both tools, the number of random factors in the data was reduced.

In order to avoid exposing our study volunteers to the violent content, we chose to focus on cyber crime, which is an important topic by itself as our Introduction and Literature Review sections indicate. Since the types of questions that cyber perpetrators are asking (e.g. "How to hack into Unix computer?", "Where do I get a password cracker?") are grammatically similar to those that "would be" terrorists may be asking (e.g. "How to build a pipe bomb?", "How to I get ammonium nitrate?") we believed this would not necessarily limit our findings to the cybercrime-related malevolent content.

Thus, we were testing the ability of the two compared tools to locate the content that may be potentially facilitating cyber criminals by providing them with instructions on how to commit illicit acts. Our volunteers placed themselves in the "hacker shoes" and came up with "indicative" questions such as "How to steal a credit card number?" etc. As the reviewed literature suggests, providing answers to those questions on a certain site may be a strong indication of carrying malevolent content. Then, we asked our volunteers to find the pages that help to answer those questions while testing which tool was more effective.

Metrics

This section discusses the two metrics that we used to operationalize our hypotheses. Ideally, the metrics should be based on the productivity in locating malevolent content, which can be measured by the time it takes to perform the tasks or the ability to accomplish the task in the time allowed [23]. However, conducting a study online would pose significant challenges in that case: subjects would need to self-

report the time spent, should not be interrupted, and trusted in their self reported measurements. Also differences in a time of a day, network traffic and hardware variability would contribute to high variance in the measurements. Thus, in this study, we involved only the analysis of the log files as described below.

Reciprocal Answer Rank

In order to evaluate the relevance of the retrieved pages we analyzed the search logs and users' judgments as detailed in the following sections. The most popular metrics to evaluate the relevance (or the likelihood to provide the answer) of the search results are those based on Recall and Precision [27]. However, it has been noticed that those metrics would not be applicable for the open domain world-wide-web searching due to the inability to obtain all potentially relevant pages. Thus, other metrics have been suggested [28] [15] [23]. By following the suggestions from the prior studies, we adopted the metric of reciprocal rank of the first web page that can qualify as containing an acceptable answer. This metric was also repeatedly used at the TREC QA competitions [36].

Users decided themselves if the page found could be considered as "containing an acceptable answer" based on our suggested criteria (listed in Appendix A.). In short, as long as the page helps to answer the question even "a little bit" it was considered a valid answer page. The users did not have to find an exact answer or to compile the answers based on multiple sources.

The reciprocal answer rank metric assigns the score of *1* to the retrieved results in which the first page provides the answer, *1/2* to the result where the second page does, *1/3* to the result where the third does, etc. The highest score means the highest relevance (likelihood of providing an answer) of the returned pages. The reciprocal transformation is needed in order to normalize the rank and avoid sensitivity to the tails. Indeed, by doing so the difference between the reciprocal ranks of *20* and *200* is much less than that of *1* and *2*. Thus, this metric is not sensitive to the exact cut-off, e.g. all retrieval results not having answer in the top *10* or *20* can be safely assigned the score of zero.

The justification behind this metric is that in a typical browsing the user scans the snippets from the top of the result list to its bottom. The users typically scan only the first *20-30* pages, with the likelihood of going further quickly decreasing. Thus, the value of the relevant page quickly decreases with its rank. We "artificially" disallowed the users to follow the links from the returned pages in order for those assumptions to be more realistic. We realized that this handicaps search engines since they may be returning good "hub" pages (those pointing to the relevant pages), however we do not believe it would affect the findings.

Each time, when the user re-formulated a Google query, we increased the rank by *10* (the number of pages returned by Google as response to a query). The assumption behind this is that the user has looked at all the top *10* snippets (or pages), did not find the answer and moved along to a different query. The situation when the user immediately decides to reformulate the query based on the other hints (such as the total number of pages matching the query or a discovered typos in the query) were not observed in our study, although admittedly could potentially violate this assumption. Shorter than *10* pages or empty lists were also a potential threat to this metric, but still never observed in our study.

Instructions and Pilot Study

Since there were practically no known prior empirical studies with Open Domain Question Answering involving human subjects in an interactive experiment, we had to come up with our own empirical design. Prior to starting our study we made our Open Domain Question Answering System publicly available on the Web at http://qa.wpcarey.asu.edu/ and announced it in various newsgroups related to the topics of AI and Natural Language Processing. We have averaged 15 trials per day since then. The QA sessions were automatically logged which allowed us to fine tune the system and also collect data on the types of questions that people may be asking. We were really surprised to see many non-factual questions such as "How to" or "How do I," which also partially stimulated our interest to this study.

Figure 1 shows a typical QA session. The interface is very straightforward: the user enters the question and receives a set of answers along with the links to the pages where the answers were found.

Insert Figure 1

Insert Figure 2

Since we only needed Google's functionality to provide keyword search, we had implemented a CGI front end to Google, limiting the user to enter only a query, thus disabling all the other potentially distracting features of the portal such as image search, tool bar icons, shopping, news, advertisement, etc. Our CGI interface redirected the query to Google without any modifications and presented the snippets returned by Google to the user, in the same order and also without any modifications. It also retained the log of the searches to be used later for analysis. Figure 2 shows our interface to Google.

For our pilot study, we asked 3 student volunteers to put themselves in the shoes of a "cyber-criminal" e.g. as if they were trying to commit some illicit actions (e.g. "hack" into computer networks) and come with 6 questions, answers to which would facilitate their intentions. In order to get familiar with the topic and get inspired, we suggested spending 10 minutes searching Google News using keywords related to cyber crime (e.g. "phishing") and skimming through the found news articles. The exact instructions are in the appendix.

Then, we asked the users to find answers to their questions using Google for 3 questions and QA tool for another 3 questions, switching turns to minimize learning effects. As discussed above, in order for the assumptions beyond our metrics to be more realistic, while searching for the answers, users were allowed to click on the links returned by the tool and read the web pages found, but were not allowed to follow the links from those pages further. For example, when Google tool found a page http://myserver.com/mypage.html which contained a link to http://myserver.com/my-referred-page.html, the user could read http://myserver.com/mypage.html but could not read http://myserver.com/my-referred-page.html. Similarly, the users could not just enter the URL into the browser and go to the other pages directly.

We videotaped one user. The other two performed the tasks online. We followed the tasks with unstructured interviews. As a result, we modified the instructions to the way presented in the appendix

and performed necessary corrective maintenance in our QA system. We also decided to impose a question length limit of 10 words to avoid long and complex questions, e.g. with conjunctions ("How to create a non-real IP identity and avoid cyber police when doing illegal stuff?") that some users initially produced.

Task Level Phase

We involved 9 volunteers in this ("task level") phase. The volunteers followed the same instructions as were discussed in the previous section. They performed their tasks online, on their own (not monitored) and at the time of their choice. The volunteers were (all but one) undergraduate students in a College Of Business in a major US research university, majoring in Computer Information Systems, familiar with web searching and the domain of investigation (cyber crime). Thus, their skill set was very similar to those typically possessed by law enforcement agents involved in cyber policing. The search logs were recorded and subsequently analyzed.

We designed our experiment being inspired by well known within information science community "gold standards" -- TREC [36] and TIPSTER [10] [11] competition-like conferences. They typically involve around 10 assessors and 50-500 topics (questions) for each task (such as QA, summarization or ad-hoc retrieval). The results are statistically analyzed using a task rather than a subject (assessor) as a unit of analysis. This is because it has been noticed that the evaluation results are more sensitive to the choice of information seeking tasks than to the number of assessor. Although the assessments of each document (answer) are known to be subjective to a certain degree, each assessor blindly evaluates several search results, thus the subjective judgments tend to "average out". The agreements among assessor also becomes much higher when used as relative evaluation of the results generated by two different tools, rather than an absolute judgment of the relevance of each specific document (answer). Although we followed the same evaluation philosophy and used a question as the unit of analysis, to be on a safer side, we verified our statistical tests (detailed below) using each subject as the unit of analysis and obtained virtually identical results.

Initially, we targeted to recruit approximately 10 volunteers for the study who would generate and evaluate approximately 30 test questions. Each user was instructed to switch the tools to use with each question in order to avoid learning effects. Indeed, if a user found an answer page using the QA tool, it would be too easy to enter several words from the answer page into Google and receive this page again among Google top search results. Giving the users the freedom to choose which tool to use was also not an option, since it could have resulted in very unbalanced data, for example if the majority of the users chose to mostly use a familiar tool (Google) or, in the opposite extreme, the tool that was offering novel experience (QA tool).

Table 2 shows the results of the Task Level phase. "NF" indicates that the answer was not found by the user within the time allowed (5 minutes), and was assigned the reciprocal rank of 0 as a result. We wanted to keep a user session within a limit of one hour, thus each user tried only 3 different questions with each tool (6 questions total). This created high variability of assessment within our sample which explains why we did not find any statistically significant difference in user preference with respect to either of the tools. There was also no statistically significant difference in the relevance of the retrieved pages as measured by the reciprocal answer rank. Thus, we were not able to conclusively answer research question Q1 (improvement at the task level). However, as we wrote earlier, we were entirely prepared for such an outcome and proceeded to the second phase which promised more statistical power due to a more efficient design.

The set of questions collected from our users confirmed our conjecture that for this application a QA system needed to handle mostly non factual questions, and thus could not be handled by the existing systems. This observation justifies our efforts to incorporate the mechanism to handle non factual types as described in the "Algorithms and Implementation" section. The "How to" type of questions was prevailing (29 out of 54). Other frequent types included "How can I" (3), "How do I" (3) and "How do you" (5), which were grammatically the same and could be automatically converted to "How to" questions. We implemented those conversions before proceeding to the next phase.

Insert Table 2

We observed that the users entered the questions directly to Google in approximately 25% of cases typically by copying and pasting from their instructions. This may indicate their expectations that the search engine would treat their inputs as questions but not just "bag of words." When asked why they did not enter their questions into Google, the users would typically say that they knew that the search engines only needed keywords and that entering only the essential (from users' perspective) words would save them some typing time. Those observations support our claim that the QA technology will likely to be accepted as an easy and natural way of information exploration, once more users believe that entering an entire question would help to locate the desired content.

Blind Evaluation Phase

This phase compared the relevance of the pages returned by the two tools. We started the blind evaluation phase while still waiting for the questions from 2 remaining users from the previous phase. We decided that the number of questions obtained from the first 7 users was appropriate for the second phase since we also wanted to keep the time spent by the volunteering users at a minimum. Since we needed the user queries from the previous phase for the comparison of the results, we limited this phase to only the questions that the users have attempted with Google; thus, we have performed the comparison on 21 questions, which we re-run in a batch mode through both tools. The same 9 volunteers involved in the previous phase were presented with the questions and the retrieved results (limited to top 10) in the same format. However, this time, the users acted as "blind" judges, since they did not know which tool produced which results.

The assigning of questions to the judges was in such a way as to minimize the impact of the random factors when considering a question as a unit of analysis. Thus, the retrieved results from both tools were assigned to the judges at random but with the following constraints: 1) Each question was assigned to at

least two judges (one with each tool). 2) No question was assigned to more than four judges (2 with each tool). 3) No judge was assigned a question that he/she contributed during the previous phase. This assignment allowed us to avoid "familiarity" effects and also distributed the questions more uniformly than with unrestricted random assignment. The instructions to locate the first answer page were essentially the same as in the previous phase.

We expected much less variability in the results compared to the previous phase because each question was run through both tools and the outputs were "blindly" evaluated. For ease of interpretation, prior to our analysis, we decided to average the reciprocal ranks across questions and compare the averages. This approach ignored possible effects by repetition of the judges or repeating the authors of the test questions. We believe those effects were negligible due to randomness of the assignments. Thus, our operatioanalized hypothesis was the following:

H1o: The QA tool produces the same reciprocal rank as Google. The alternative hypothesis *H1a* was that the QA tool performed better.

Table 3 shows the evaluation results for each question. We rejected both H1o at the levels of confidence a < 0.1 (p-value = 0.08), thus empirically obtaining confirmation that the overall relevance of the returned pages was better with the QA tool and answering positively our second research question. The relative improvement was quite substantial: by using Question Answering technology, the average reciprocal ranks were increased by up to 25%, which we believe is a practically important result.

We have observed that the following major categories of web pages were indicated by our users as providing the answers to their questions (in the approximate order by their frequency of occurrence), and thus potentially may contain malevolent content:

1) Message boards, newsgroups or discussion forums. Those sites typically allow unrestricted posting of messages, anonymous or using an alias. Very often, the answer found is a post, titled as a response to another post, e.g. "Re: I wanna crack hotmail passwords."

2) Standalone web pages that may be considered "hacking" tutorials, guidance, advice etc. They frequently have "How to" in the title or headings. Many of the web sites hosting them have international (non US) domain names.
3) Technical documentation from software vendors. Those pages would be typically posted for cyber security experts or just general computer users, and serve completely legitimate purposes of educating on the issues of security, e.g. for the purpose of avoiding fraud or security breaches. However, many potential intruders can get "educated" from those pages as well.

To preserve the anonymity of the owners of the websites and the authors of the posts we decided not to mention any specific URLs or domain names in our paper.

Insert Table 3

Conclusions, Limitations, and Future Research Directions

We have established that pattern based Question Answering technology is more effective at locating web pages that may provide answers to the set of indicative questions (such as "How do I crack passwords?", "How do I steal a credit card number?" etc.). The pages providing answers to those questions very frequently contain malevolent content and co-exist with illicit online activities (cyber-crime). We suggest that the QA approach is superior than keyword searching since the latter locates many innocuous pages (news, discussion forums) that use the same words and phrases as the malevolent pages, which results in information overload. QA technology narrows down search results to precise answers and "how to" manuals -- the types of content more likely to be abetting cyber criminals. Thus, QA can be used in addition to traditional keyword searching by law enforcement, public watchdogs and researchers who are trying to locate and monitor malevolent online content. We would like to note that other (non-related to this study) techniques to narrow down search results have been suggested, e.g. those based on recognizing document genre [23] or focused crawling [4].

Using QA technology can also reduce cognitive load during regular monitoring when repeated queries need to be run and the retrieved pages re-analyzed. While it is extremely cumbersome and tedious for

human users to look for new sites and posts, re-running automated QA in background offers much more efficient solution. We believe that our approach can be extended to other applications of Information Technology where an organization can benefit from monitoring the dynamic environment in which it exists. The specific examples can include Business Intelligence [1] [25] and Knowledge Management [5]. The approach can be used in tandem with automated summarization techniques, e.g. those that can summarize the changes in the search results [23]. Instead of dealing with the flood of possibly repeating content, an analyst will be able to quickly glance through the fresh information "nuggets" and notice the important trends.

As a by product, we have compiled a collection of "indicative" questions to which potential cyber violators may be seeking answers and concluded that existing Web QA systems mentioned in our Literature Review are not able to locate pages that contain the answers (and thus be potentially malevolent) since they are suited only for factual, definition or list questions. For this reason, we have advanced pattern based Question Answering technology by adding the ability to locate the answers to non factual questions, such as "How to plant a computer virus?" , "How do I break into Unix server?" etc.

Our direct practical implications are to law enforcement officers and to the law-enforcement IT systems designers. Our results suggest that in addition to the popular keyword search it pays off to invest in the tools that are based on Question Answering technologies, specifically those similar to our pattern based probabilistic approach. QA approach to locate malevolent content is also more intuitive and natural than the traditional keyword approach since the interaction is based on natural language questions, which are easier to form than keyword queries, especially for novices in searching technology. This also allows us to believe that acceptance of QA technology will not present a significant problem.

We focused our efforts on the technical aspect of the problem and deliberately have not considered any ethical or political aspects, for example the legal and political feasibility of monitoring or removing malevolent content from the web and prosecuting its authors and hosts. The literature reviewed in our paper suggests that such actions are possible and carried out in practice.

We realize that QA technology can be also abused if fallen into wrong hands, as any other advanced technology can, and help cyber criminals to locate the content of interest. However, considering that this social group is typically computer savvy and have adequate time at their disposal, we do not think having access to QA technology would make a substantial impact on that group. The potential violators currently have no problems locating the content by keywords or going directly to the resources (such as web sites, newsgroups, forums, etc.) referred by peers or from the other resources already known to them.

The researched QA technology can also be used by security experts for educational purposes or as a complement to online lectures on cyber security.

There are a number of methodological limitations already mentioned throughout our paper, which we hope to resolve in the follow up studies. For example, we would like to test separately the effects of the components involved in our QA system, specifically: triangulation, pattern and heading matching, or training the pattern weights. Although we have tested all types of questions altogether, the exact numerical comparison may happen to be sensitive to the specific types of questions, e.g. factual vs. non-factual ("how to" vs. "what is").

This is the first empirical study involving an interactive web based open domain question answering system and the first comparison against keyword searching as a baseline. We deliberately did not embed any domain specific decisions while designing and implementing our QA system, so from a technological perspective, it remains an open domain system. This allows us to believe that our results may be extended to the open domain question answering in general, including many other important applications where locating online content quickly is vital. This includes business intelligence, intellectual property protection, digital forensic and preventing acts of terrorism. Investing into new emerging technologies, as the one studied here, makes our world safer and more prosperous.

Acknowledgements

We would like to thank the following people familiar with cyber crime and cyber terrorism for their advice: Robert A. Ellison, from KGHS Consulting (a former Supervisory Special Agent of the Federal Bureau of Investigation); Ellis Chip from National Memorial Institute for the Prevention of Terrorism and

Edna Reid, a Research Scientist from the University of Arizona (a former agent of the Central Intelligence Agency).

References

[1] Bernstein, A., Grosof, B. and Provost, F. Business Intelligence: The Next Frontier for Information Systems Research? Panel Description, in Proceedings of the Workshop on Information Technologies and Systems (New Orleans, LA, USA, December 15-16. 2001).

[2] Brin, S., and Page, L. The Anatomy of a Large Scale Hypertextual Web Search Engine. Stanford technical report. http://dbpubs.stanford.edu:8090/pub/showDoc.Fulltext?lang=en&doc=1998-8&format=pdf&compression= (1998).

[3] Burke, R., Hammond, K. and Kozlovsky, J. Knowledge-based information retrieval for semi-structured text. In AAAI Fall Symposium on AI Applications in Knowledge Navigation and Retrieval, pp. 19–24, (1995).

[4] Chakrabarti, S., van den Berg, M., Dom, B. Focused Crawling: A New Approach to Topic-Specific Web Resource Discovery, in: Proceedings of 8th World Wide Web Conference (Toronto, Canada, May 11-14, 1999).

[5] Chen, H. Knowledge Management Systems: A Text Mining Perspective, University of Arizona (Tucson, Arizona, November, 2001).

[6] Chen, H. The Terrorism Knowledge Portal: Advanced Methodologies for Collecting and Analyzing Information from the Dark Web and Terrorism Research Resources, presented at the Sandia National Laboratories (August 14, 2003).

[7] Claburn, T. Search For Tomorrow, Information Week (March 28, 2005).

[8] Dumais, S., Banko, M., Brill, E., Lin, J., and Ng, A. Web Question Answering: Is More Always Better?, in: Proceedings of ACM Conference on Information Retrieval (2002).

[9] Harabagiu, S., Moldovan, D., Pasca, M., Mihalcea, R., Surdeanu, M., Bunescu, R., Girju, R., Rus, V., & Morarescu, P. Falcon: Boosting knowledge for answer engines, in: NIST Special Publication 500-249: The Ninth Text REtrieval Conference (TREC 9), pp. 479–488. (2000).

[10] Harman, D. The DARPA TIPSTER Project, in: SIGIR Forum, 26(2), pp. 26-28. (1992).

[11] Harman, D. Proceedings of the TIPSTER Text Program Phase I (San Mateo, California, USA. 1994).

[12] Hasday, J. L. Columbine High School Shooting: Student Violence (American Disasters) (Enslow Publishers, July 1, 2002).

[13] Jacques, R. 28 Arrested in Global Web Fraud Sting, in: E-commerce times. (October 2004. Available at http://www.ecommercetimes.com/story/37718.html)

[14] Lempert, R. J., Popper, S. W., Bankes, S. C. Shaping the next one hundred years: new methods for quantitative, long-term policy analysis (RAND, Santa Monica, CA. 2003).

[15] Leouski, A., & Allan, J. Visual Interactions with a Multidimensional Ranked List, in: Proceedings of the Twenty First Annual International ACM Conference on Research and Development in Information Retrieval, pp. 353-354 (Melbourne, Australia, 1998).

[16] Lyman, P., and Varian, H.R. How Much Information, (2000). Retrieved from http://www.sims.berkeley.edu/how-much-info on 11/4/2004.

[17] National Science Foundation. NSF Announces $30 Million Program in "Cyber Trust." (2003).Available at http://www.nsf.gov/od/lpa/news/03/pr03133.htm

[18] Powell, W. Anarchist Cookbook. (Ozark Pr Llc, 1970).; Reissue edition, September 2003.

[19] Radev, D. R., Libner, K., & Fan, W. Getting answers to natural language queries on the web, Journal of the American Society for Information Science and Technology (JASIST), 53(5) (2001).

[20] Radev, D., Fan, W., Qi, H., Wu, H., Grewal, A. Probabilistic question answering on the web, in: Proceedings of the 11th WWW conference (Hawaii, 2002).

[21] Radev, D., Fan, W., Qi, H., Wu, H., Grewal, A. Probabilistic question answering on the web, Journal of the American Society for Information Science and Technology (JASIST), 56(6) (2005).

[22] Reid, E., Qin, J., Chung, W., Xu, J., Zhou, Y., Schumaker, R., Sageman, M., Chen, H. Terrorism Knowledge Discovery Project: A Knowledge Discovery Approach to Addressing the Threats of Terrorism, in: Proceedings of the Second Symposium on Intelligence and Security Informatics, (Tucson, AZ, June 10-11 2004).

[23] Roussinov, D., & Chen, H. Information navigation on the web by clustering and summarizing query results, Information Processing and Management, 37 (6) (2001).

[24] Roussinov, D., and Robles-Flores, J. Web Question Answering: Technology and Business Applications, in: Proceedings of 2004 Americas Conference on Information Systems (August 6 – 8, New York, NY, 2004).

[25] Roussinov, D., and Robles-Flores, J. Web Question Answering: Technology and Applications to Business Intelligence, International Journal of Internet and Enterprise Management, 3(1) (2005).

[26] Roussinov, D., Crowston, K., Nilan, M., Kwasnik, B., Cai, J. and Liu, X. Genre Based Navigation on the Web," in: Proceedings of Hawaii International Conference on System Sciences (HICSS-34), (Island of Maui. January 4-7, 2001).

[27] Salton, G. and McGill, M.J. Introduction to Modern Information Retrieval (McGraw-Hill, New York, 1983).

[28] Shneiderman, B., Byrd, D., & Croft, W.B. Sorting out searching: A user-interface framework for text searches, Communications of the ACM, 41(4) (1998).

[29] Shneiderman, B., Byrd, D., and Croft, W.B. Clarifying Search: A User-Interface Framework for Text Searches. DLib Magazine (1997).

[30] Soubbotin, M., & Soubbotin, S. Use of patterns for detection of likely answer strings: A systematic approach, in Proceeding of TREC (2002).

[31] Surdeanu, M., Moldovan, D. and Harabagiu, S. Performance Analysis of a Distributed Question Answering System, IEEE Transactions on Parallel and Distributed Systems (June, 2002).

[32] Swartz, J. Hackers hijack federal computers, USA Today, (August, 2004). Available at http://www.usatoday.com/tech/news/computersecurity/2004-08-30-cyber-crime_x.htm

[33] Turetken, O., Sharda, R. Development Of A Fisheye-Based Information Search Processing Aid (FISPA) For Managing Information Overload In The Web Environment, Decision Support Systems 37(3) (2004),

[34] Vasagar, J. Deadly net Terror websites easy to access. The Guardian (Saturday July 1, 2000). Available at http://www.guardian.co.uk/bombs/Story/0,2763,338617,00.html

[35] Verton, D. and Verton, D. Black Ice: The Invisible Threat of Cyber-Terrorism (McGraw-Hill Osborne Media, 19 August, 2003).

[36] Voorhees, E. and Harman, D., Eds. Proceedings of the Tenth Text REtrieval Conference TREC (2004).

[37] Voorhees, E. and Buckland, L., Eds. Proceedings of the Twelfth Text REtrieval Conference TREC, (Gaithersburg, Maryland, USA, November 18-21, 2003).

[38] Weimann, G. How Modern Terrorism Uses the Internet. SPECIAL REPORT 116, United States Institute of Peace, (2004). Available at http://www.usip.org/pubs/specialreports/sr116.html.

[39] Wells, H.G., The Time Machine, (Tor Books, 1895) Reissue edition (December 1, 1995).

Appendix A. Instructions for the Study

This study tests applicability of the modern Question Answering Technologies to locating content on the World Wide Web that can be potentially used by Cyber criminals. We need you to help us create a set of test questions related to cyber crime. For that, we will ask you to put yourself in the shoes of a cyber-criminal e.g. as if you were trying to be a malicious hacker planning "attacks" on computer networks. In order to get familiar with the topic and get inspired, we suggest that you search Google News for something related to cyber crime and skim through one or two news articles of your choice. Please allow yourself 10 minutes for that task, and then go to the next paragraph.

Now, we ask you that you write 6 questions that a potential Cyber violator may be asking. Please try to come up with specific questions, for example: "How to break into a Linux system?" or "How to scan a computer port." Each question should be **no more than 10 words long**. Remember, this task is a brainstorming exercise, so please be creative!

Searching Using Google

Please go to the link ([removed for blind review]) and search for the answers to your questions **1, 3, 5.** This tool (called Google for the purpose of this experiment only) can take queries the same way as Google (or other similar ones) search engine does. Please do not use any search engines (like Google, AltaVista, etc.) directly! While searching, please use the following guidelines:

For each question, you need to find only the first link in the order presented that point

to a page that you think would help you to answer your question. **IMPORTANT**: this page does not have to provide you with the complete answer. As long as you think it helps you to answer your question even a little bit you should copy and paste it below and finish your task. Again, this is not a test! There are no right or wrong answers. Just use your own judgment. If **after 5 minutes** you cannot find an appropriate page, just type "not found" and continue to the next question.

While searching for the answers, you can click on the links returned by the tool and read the web pages found by either of the tools. But you can not follow the links from those pages further. Example: Google tool finds a page http://myserver.com/mypage.html which contains a link to http://myserver.com/my-referred-page.html You can read http://myserver.com/mypage.html but you can not read http://myserver.com/my-referred-page.html. Similarly, you can not just enter URL into your browser and go to the pages not found by the tool.

Searching Using QA tool

Now, you will use the **QA tool**, tool that was specifically designed to find the answers to natural language questions on the Web. The tool is available at http://qa.wpcarey.asu.edu. While searching for the answers, please copy and paste your questions into the tool. The tool may take up to 1-2 minutes to find the answers, so please be patient!

For each of your questions 2, 4, and 6, using the same guidelines as earlier with Google (listed on page X) please find the first page that helps answering your question.

how to \Q * Q&A Below are some ways in which to \Q * Below are some ways to \Q * recipe to \Q * software to \Q * software that can \Q * tool that can \Q * tool to \Q * * can \Q * * could \Q * * can \Q via * * has the ability to \Q * how does someone \Q * anyone can \Q * someone can \Q *	* to \Q you * How to \Q 101 * Re : how to \Q * Re : * \Q * the easiest way to \Q * the * way to \Q * The way to \Q * * to \Q you need to * * to \Q just * * to \Q use * * FAQ on how to \Q * to \Q requires * How to \Q \p Use * How to \Q \p You need * How to \Q \p The way to do it is *

Table 1. Answer pattems used for "How to" questions. \Q is the "Question Part", e.g. "create a virus.

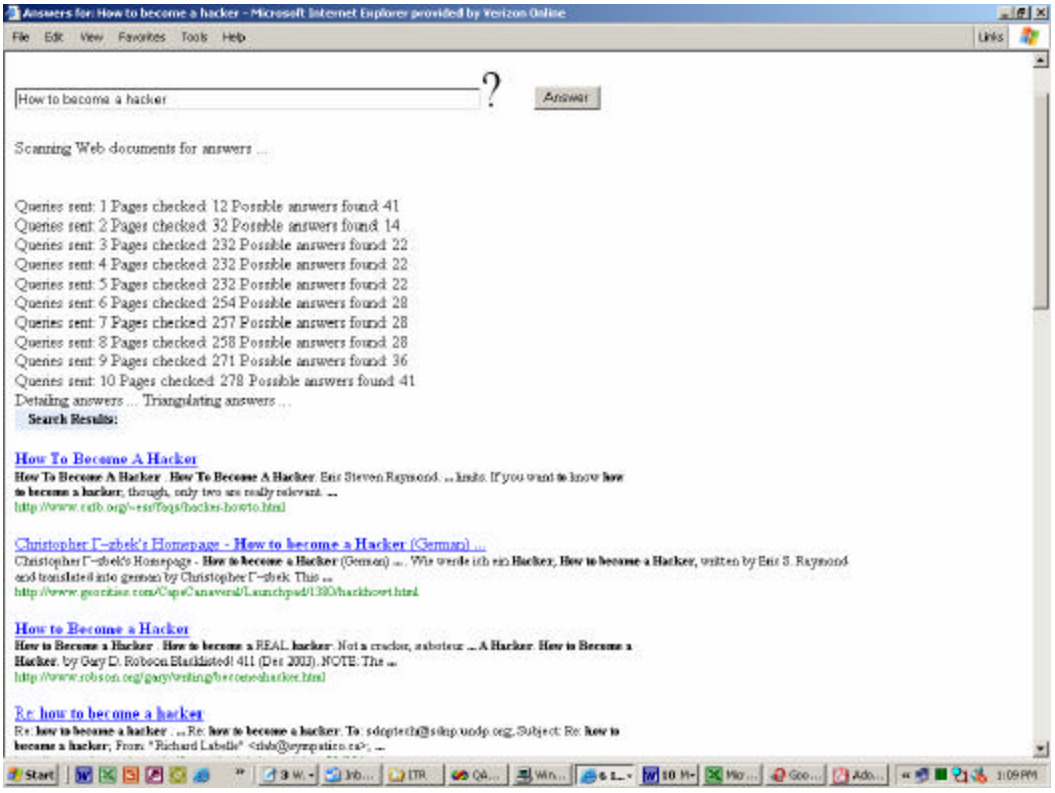

Figure 1. Question Answering session.

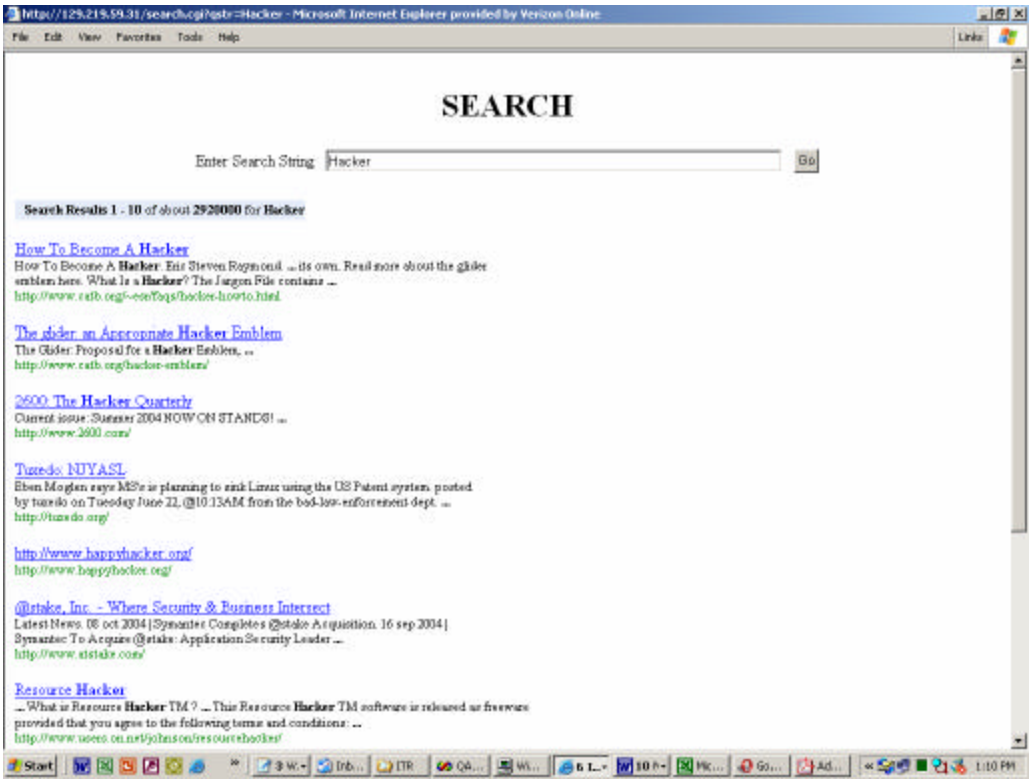

Figure 2. Front end to Google.

User	Question	Google Answer Rank	QA Answer Rank
1	How to use e-mail to get people's bank information?	41	
	How to install a Trojan into another computer?		4
	How to bring down a website temporarily?	NF	
	How to access restricted government information?		NF
	How to steal Microsoft software?	NF	
	How to infect a computer using Microsoft Outlook?		2
2	How to spoof a web server?	NF	
	How do you google passwords?		2
	What are some password dictionaries?	32	
	What are some recent Internet Explorer security flaws or vulnerabilities?		5
	How do you generate credit card numbers?	NF	
	What are recent SQL Server injection attacks or vulnerabilities?		NF
3	How many feet do I need to be from a wireless	3	

	access point in order to piggy back on it?		
	Which scripting language is Windows XP most vulnerable to?		2
	What are the latest Internet Explorer security updates?	1	
	How do I create email spam?		5
	What is SpyBot?	1	
	How do I hack NAT?		NF
4	What is the default admin password for Windows XP?	1	
	How to crack yahoo mail passwords?		1
	How to crack hotmail email passwords?	1	
	How do you remotely access a computer?		1
	How do you edit someone's registry?	11	
	How do you bypass a firewall?		1
5	How to crack encryption keys?	1	
	How to exploit active x?		6
	How to create distributed denial of service attacks?	4	
	How to phish credit card numbers?		1
	How to exploit buffer overflow?	4	
	How to spoof e-mail addresses?		15
6	How to create a Trojan Horse?	4	
	How to gain access in a protected WIFI network?		5
	How to create a virus?	2	
	What is a common entry port for a government system?		NF
	How to embed a worm into an e-mail?	2	
	What is a common entry port for a corporation?		8
7	How to spread computer virus?	12	
	How to steal identity?		NF
	How does 'phishing' work?	4	
	How to hack bank encryption system?		6
	How to identify unprotected wireless network?	NF	
	How to break into computer network?		3
8	What language would I use to program a virus?	40	
	What strategy can I use to circumvent a corporate firewall?		3
	How can I take advantage of vulnerable wireless networks?	1	
	How can I gain access to a corporate database remotely?		6
	How can I create a denial of service attack?	4	
	How do I make a Trojan horse program?		NF
9	How to create a fake shopping website?	NF	
	How to conduct a DoS attack?		10
	How does buffer overflow work?	1	
	How to capture the password sent in the network?		3
	How to acquire UNIX shadow file?	NF	
	How to create a spyware?		5

	Mean Reciprocal Rank	0.36	0.33
	Standard Deviation of the Mean	0.14	0.11

Table 2. The results from the Task Level Phase.

User	Question Number	Reciprocal Answer Rank QA	Reciprocal Answer Rank Google
1	1	0.25	0.26
	3	1	0.2
	5	0.16	1
2	1	0.6	1
	3	0.75	.33
	5	0.25	
3	1	1	0
	3		
	5	0.25	0
4	1		0.5
	3		0.05
	5		
5	1	0.14	0.05
	3	0.75	0.5
	5	0.25	0.33
6	1	0.25	0.25
	3	1	0.42
	5	1	0.5
7	1	0.42	
	3	0.5	
	5		1
Mean:		**0.51**	**0.41**
Mean Standard Deviation		**0.027**	**0.029**

Table 3. Blind relevance judgments.

Biographical Notes:

Dr. Dmitri Roussinov is currently an Assistant Professor in the Department of Information Systems, at Arizona State University. He has published several articles in leading information systems and science journals, such as *Decision Support Systems,* and *Information Processing & Management.* He received his Ph.D. in Management Information Systems from University of Arizona and has an MA degree in Economics from Indiana University, and a diploma with honors in Computer Science from Moscow Institute of Physics and Technology, in Russia. Prior to joining ASU, Dr. Roussinov served on the faculty of Syracuse University, School of Information Studies for two years. His recent studies involved clustering of text documents, mining semantic similarity relationships from co-occurrence statistics, identifying the genre of web documents, machine learning, and question answering on the Web.

José A. Robles-Flores is a PhD student in the Department of Information Systems at the W.P. Carey School of Business, Arizona State University. He holds a B.S. degree in Computer Science from Francisco Marroquin University in Guatemala and a Masters degree in Business from ESAN, in Lima, Peru. He has gained experience in software engineering and Internet project management, while working for the Peruvian Tax Administration and the Peruvian Scientific Network (first non profit ISP in Peru). He is an instructor at ESAN University in Lima, Peru. Jose is experienced in teaching information systems courses, such as databases, information technology for businesses, knowledge management, and information resource management. His research interest is in the areas of Information Retrieval and Knowledge Management.