Directing Web Search Engines using a Knowledge Amplification by Structured Expert Randomization Architecture

S.H. Rubin[1], Isaí Michel Lombera[2], Michael Armella[3], Jeremy Conn[3], S.C. Chen[3] and G. Lee[2]

[1]SPAWAR - Systems Center
53560 Hull Street
San Diego, CA
92152-5001

[2]Dept. of Electrical and Computer Engineering
San Diego State Univ.
San Diego, CA 92182

[3]School of Computing and Information Science
Florida Int'l University
ECS 362
Miami, FL 33199

Abstract. The capability to dynamically retrieve detailed multimedia which may come from knowledge bases as well as sensor information in response to specific user queries offers the potential to create decision support systems of unprecedented utility. Such systems can learn from user feedback; by minimizing the system training required of the knowledge engineer, we can more effectively process vast free-text databases of knowledge for minimal development cost. Furthermore, these bases may be concurrently created and maintained and search algorithms can run on parallel processors connected in heterogeneous distributed networks. This paper presents an approach to web searching using knowledge amplification by structured expert randomization platform. Creativity, training, and decision support via just-in-time knowledge are fused in a system that provides the intelligence analyst with critical information where and when it is needed and in a user-friendly format. In this paper, national security examples are selected; extension to a plethora of other domains for decision support and training is inevitable.

1 Web Search Engines

The World Wide Web continues to play an important role in storing and retrieving informational databases for many applications in the civilian, commercial and military environments. Search engines for the general web typically do not really search the World Wide Web directly. Each one may search a database of the full text of web pages automatically harvested from the billions of web pages residing on servers. When one searches the web using a search engine, one may be searching a somewhat stale copy of the real web page. When one clicks on links provided in a search engine's search results, usually one retrieves the current version of the page from the server.

As is well-known, search engine databases are selected and built by computer programs denoted as spiders. These programs "crawl" through the web, finding pages for potential inclusion by following the links in the pages they already have in their databases (i.e., already "known about"). The programs can do not "think" or use judgment to "decide to go look something up and see what's on the web about it".

After spiders find pages, they pass them on to another computer program for "indexing". This program identifies the text, links, and other content in the page and stores it in the search engine database's files so that the database can be searched by keyword and whatever more advanced approaches are offered, and the page will be found if the search matches its content.

There is a need to provide intelligence in web searching. Note that web searching can be used to poll sensor network information whereby sensor data can be fed to the user through a web interface. In order for the search engine as well as the user to *learn* as one searches the web for information, a knowledge amplification architecture is suggested to accelerate the learning process thereby improving the efficiency of the search. This synergy between user and the knowledge base with the knowledge amplifier as the conduit accelerates the learning process. The knowledge amplifier is now discussed.

2 An Introduction to the KASER Algorithm

The KASER is a knowledge amplifier (the acronym stands for *K*nowledge *A*mplification by *S*tructural *E*xpert *R*andomization) based on the principle of randomization. This principle refers to the use of fundamental knowledge in the capture and reduction of a larger, dependent space of knowledge (not excluding self-reference). In a KASER system, the user supplies declarative knowledge in the form of a semantic tree using single inheritance. Unlike conventional intelligent systems, however, KASERs are capable of accelerated learning in symmetric domains [1-2].

Conventional expert systems generate cost curves below the breakeven line. In conventional expert systems, cost increases with scale and the increase is never better than linear. In the case of KASER systems, the cost *decreases* with scale and is always better than linear, unless the domain is asymmetric (random). Perfectly (asymmetric) random domains are trivial constructs and

are not encountered in the construction of practical applications [3].

Conversely, perfectly symmetric (non-random) domains are also trivial and are also not found in practice [3]. In other words, a perfectly random domain would have no embedded patterns (true random numbers), while a perfectly symmetric domain would be infinitely compressible (free of information content). Clearly, such constructs are strictly artificial. The more symmetric is the operational domain, the less the cost of knowledge acquisition.

As a synopsis of the KASER, a production rule is defined to be an ordered pair whose first member is a set of antecedent predicates and whose second member is an ordered list of consequent predicates. Predicates can be numbers or words [4-5]. The linking of the two members forms rules or courses of action.

KASER systems can be classified as Type I and Type II, depending on their characteristics. In a Type I KASER, words and phrases are entered through the pull-down menus. The user is not allowed to enter new words or phrases if an equivalent semantics already exists in the menu. In a Type II KASER, distinct syntax may be equated to yield the equivalent normalized semantics. The idea in a Type II KASER is to ameliorate the inconvenience of using a data entry menu with scale. In a Type II KASER, selection lists are replaced with semantic equations from which the list problem is automatically solved.

Thus a KASER system can amplify a knowledge base. It represents an advance in the design of intelligent systems because of its capability for symbolic learning and qualitative fuzziness. In a conventional expert system, the context may cover the candidate rule antecedent, in which case an agenda mechanism is used to decide which matched rule to fire (most-specific match, first to match, chance match.). The KASER system follows the same rule-firing principle – only the pattern-matching algorithm is necessarily more complex and embeds the conventional approach as its degenerate case.

In order to transmit and receive information back and forth between the user, the web and the KASER system in a symbiotic manner, a novel graphics-user-interface has been designed. We note that the GUI plays an important role in supporting learning for the KASER through the user. In fact, this relationship accelerates learning through visualization.

3 The Web Searching Design: The Multimedia Component and Integration with the KASER

Consider the case when multiple sensors acquire information that needs to be fused in some synergistic manner. In this paper, we consider the web as a feasible method to acquire and fuse such information. Further, this information can be prioritized, based upon how the user values the data acquired. The applications are many as one could also apply the architecture for web browsing to gather information from databases attached to the internet through web links. To interface the KASER with the user, with the intent of performing web searches, a multimedia architecture is envisioned (see Figure 1). Multimedia objects are defined to be text blocks, video snippets, slide shows, and/or sound bytes. Databases may also come from sensors such as images, acoustic detectors, or infrared devices which need to be integrated in a web based format. The shorter these objects, the more reusable generally. Thus, videos will be of a pedagogical nature and average about 15 seconds to one minute in length. For example, a video might explain complex chemical procedures for the identification of chemical-biological threats to the water supply (e.g., Keith [6]).

First, the user supplies a query. The query is automatically supplied to the Web Search box (i.e., as determined by a checkbox on the user settings page) to search for all of the most relevant instructional/decision support materials. Meta-search engines are no longer recommended. Web search engines will effectively filter out, or take into account, such words as “how”, “can”, “I”, “the”, “against”, “a”, etc. from the initial query. We note here that web search engines are sequence sensitive. ‘OR’ searches are used for variants (e.g., airline OR plane). Capitalization does not matter – excepting in the specification of the disjunct, ‘OR’.

In addition to query spell checking (or auto-correction as one types/speaks), queries will be dynamically maintained in an auto-associative list. Then, the system will offer to extrapolate the user’s prefix upon its recognition. This capability for auto-completion will also serve to improve the utility of a speech-recognition system front end.

The initial query will delimit the returned results from no data found to many pieces of data found. Pages may embed multimedia links, contain pdfs, Word, Excel, PowerPoint, and/or sensor data formats. A checkbox on the user-settings page will be used to determine which formats to pull the textual content off of or ignore (e.g., scan all .pdf files or ignore all .avi files). If too few or irrelevant pages (i.e., based on a defined threshold on the user-settings page) are returned, as gathered from the counter on the source page for the search engine(s), then the user may rephrase the query and/or modify the content of the S textboxes. In addition, if too many pages are returned by the query, then the user may scan only the first *m* websites and/or fill the first *n* pages with filtered text and/or stop upon timeout – as defined on the user-settings page.

Embedded links to multimedia will be shown in thumbnail form at the four corners of the screen. In addition, the user may define templates or schema (along with simple algorithms that select which schema to apply when) to use for this or similar displays, which may be accessed through the user-settings page.

A textbox will allow for the inclusion of embedded Web text and/or media links or sensor data. Multimedia can otherwise be sorted for relevancy and presented in sequence. It can be gathered from all relevant web pages/paragraphs having the highest possibilities.

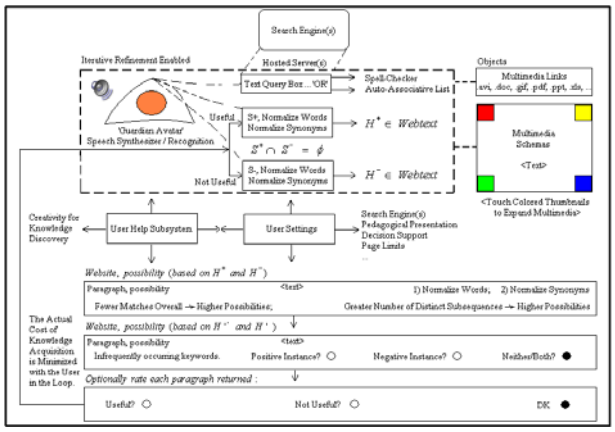

Fig. 1. GIU Diagram of the KASER-based Web Search System

The KASER is designed to return a sequence of paragraphs, which are literally retrieved from the first *m* websites (or sensor nodes which interface to the internet), produced in response to the initial query. In this initial work, we concentrate on web pages and text. The web pages are normalized by removing html format instructions so that each page will be presented using a uniform simple textual format. Each paragraph is tagged with a numeric *possibility* and (categorically by website) returned in non-increasing order of this relevancy statistic. The user sets a squelch through the user-settings page, which filters out paragraphs (websites) that have a possibility below threshold, which is zero by default. This squelch can be set to return some percentage of the best of the remainder, or alternatively up to some fixed number of paragraphs and/or pages – again through the use of parameters set on the user-settings page. Duplicate websites/paragraphs are checked for and are never written.

In addition to an interactive query textbox by way of a user-defined avatar, there are two additional textboxes: The S+ textbox allows the user to enter descriptive sentences pertaining to what they are looking for and may be arbitrarily long. The S- textbox allows the user to enter descriptive sentences pertaining to what they do not want to retrieve. The use of these positive and negative textboxes, respectively, is optional and if left blank, the system will function much as does the underpinning web search engine(s).

Again, all communications between the user and the web-search engine(s) are performed by way of a user-defined avatar. Speech synthesis may be incorporated in the future for interacting with an avatar and/or for reading text (e.g., for the visually impaired) as set by a parameter on the user-settings page. The system will be able to work with speech synthesizers and avatars, but will be constructed using high-quality commercial products (COTS) as appropriate. Ideally, but not necessarily, the user will be able to switch between "freeware" and COTS through the user-settings page. Moreover, the eventual availability of quality affordable speech recognition software will allow queries (and their replies) to be ported to any number of handheld devices (e.g., cell phones).

Let, S+ be the sequence of words found in the positive textbox and let S- be the sequence of words found in the negative textbox, where S denotes both generically. It is further required that S+ $\cap$ S- be the empty set. For example, the word, "the" (and others) are removed from S because they are common to both S+ and S- and thus cannot be used to discriminate the positive from the negative content. These words are not deleted from S+ or S-. Rather, if hashing a word or sequence of words for placement in one hash table finds it to be already present in the inverse hash table, then rather than place the word or sequence, it is expunged from the hash table where it was found.

A hash dictionary may be incorporated to normalize all words – excepting the query box, but including S and the web text to be crawled. For example, "crawled" is replaced by "crawl" for prefix-based pattern-matching purposes (i.e., "crawl" then matches "crawled", "crawling", but not "crawler"). Similarly, "commonly" is replaced by "common"; "statistics" is replaced by "statistic", and so on. However, words such as "heater" may not be replaced by "heat" though "laser" may be replaced by "lase", but "ruler" not by "rule". The rules for grammatical randomization may thus be manually defined by linguistics professionals in a spiral development. Note that the user is never shown the transformed web text – only the text as it was prior to any transformations. The transformed web text is only used for pattern-matching purposes.

Moreover, carefully selected synonyms will also be normalized, such as ship ← {boat, vessel}. The words in the set on the right must be in normalized form, as previously described (e.g., vessel ← vessels). Thus, two successive hashes are required using two distinct hash tables (i.e., the first for the normalization of words, if any, and the second for the reduction of synonyms, if any). Care is exercised so that cyclic reductions (infinite loops) do not occur.

The solution to normalized pattern-matching applies the fact that more text is statistically more likely to embody more synonyms than is less text. This in turn implies that S be written very comprehensively. Moreover, a feature allowing the user to iteratively reduce the text returned (i.e., after user review) through the specification and extension of S would serve to guide the

user in iteratively writing an ever-more comprehensive S – allowing for user-deletions from S too. It is possible to associate the additional text with its parent (i.e., machine learning). However, this is not done because the space of possible queries is so large relative to the number of training instances here.

Two separate additional hash tables are created, H+ and H-, for S+ and S-, respectively, and they are populated with normalized words and phrases (not to extend across sentence boundaries) found in the normalized positive and negative textboxes. The design is to crawl through each returned web page (up to the number of pages retrieved or produced as constrained by the user-settings page). Sentences or phrases are delimited by colons, semicolons, dashes, and periods followed by a space. In this manner, such *words* as pre-cognate or 3.1415926 do not act as delimiters.

In order of non-decreasing length, we set the frequency count for all sequences in H, for which there exists an embedded subsequence having a frequency count of zero, to zero. The system eliminates all sequences from H for which the frequency count is zero. This methodology allows for the rapid elimination of sequences that have no chance of finding a match in the normalized web text.

Here, the frequency count for each entry in H reflects the number of distinct paragraphs that the normalized word, phrase, or statement (referred to as a *sequence*) occurs in and is termed, the entries *actual frequency count* (AFC). That is, two or more occurrences in the same paragraph are only counted once. This is because subject declarations are qualitative – not quantitative.

Next, the system computes the possibility for each paragraph as follows. We check each member of H for occurrence in the current normalized and period-delimited paragraph and obtain its AFC from H to associate with the current paragraph. Again, we do not count the same subsequence twice for the same paragraph – even should it occur more than once.

If the user-settings page specifies a pedagogical presentation, then each candidate webpage will have the possibilities for each of its constituent paragraphs summed and divided by its total number of paragraphs. This ranks each candidate web site – each of which (along with its contents) is presented contiguously for the user in order of non-increasing average possibility for the *website* (e.g., for many types of pedagogical presentations).

Similarly, if the user-settings page specifies a decision support presentation and the possibility for a paragraph is below the user-set threshold, or negative by default, then the paragraph is removed from the reply. Only paragraphs that are above threshold are presented in non-increasing order of their possibility without regard for the website, from which they were taken (e.g., for many specific decision support tasks).

The normalized websites above a threshold are saved on the client for post-processing to further constrain them prior to presenting to the user for adjudication. Only normalized keywords (typically nouns) and not longer sequences are considered here because the user cannot afford the time to manually review the less important sequences. Here, word frequencies are computed over the remaining filtered websites just as was previously done for S+ and S- (e.g., counting a word at most once per occurrence per paragraph) as follows to arrive at a new H'. The maximum number of such presentations is determined on the user-settings page. These keywords are presented to (and enunciated for) the user in order of non-decreasing frequency of occurrence up to some user-defined count and/or threshold frequency as determined on the user-settings page. Pre-processing using S has previously delimited the user task, while using H' here provides additional focus for the user (e.g., in rating the utility of the returned material). The user then says, clicks, and/or touches, using checkboxes, whether the presented word is a positive instance, a negative instance, or neither/both (by default). The possibility for each paragraph is computed as before and is presented to the user in stable (i.e., preserving the previous ordering where possible) rank order in accordance with the presentation options found on the user settings page.

Each paragraph (i.e., never by website) returned and rank ordered by this process is tagged with a radio button set to a third-choice default of “DK”, which can otherwise be set to “Useful” or “Not Useful” to provide the KASER with simple, but valuable feedback. When a user reads a paragraph produced in response to their query, they optionally rate it. This informative feedback is used to support depth and creativity in queries as follows.

Paragraphs that were rated by the user as “Useful” are taken in union in their presented sequence as are those that were rated as “Not Useful”. Both groups are cached on the client – up to the content limits, if any, found on the user-settings page. Members of the former group (having the most positive possibility) serve as the new S+ and the latter (having the most negative possibility) as the new S-. Members having a zero possibility have no bias and are thus not included. The process may be iterated indefinitely (at user definition), or until a fixed point is detected, for each new set of websites so found. This serves to inject creativity into the process of knowledge discovery for training and/or decision support.

The KASER makes better use of existing knowledge/data bases and the human in the loop. For example, in the domain of chess, the query as to whether or not to exchange a knight for a rook, where the knight simultaneously attacks the queen, might produce a paragraph (with supporting multimedia), pertaining to the principle of forking in chess. The user would then be empowered with just-in-time knowledge that is symmetric with previously acquired and understood knowledge (i.e.,

otherwise the user would not be able to apply it) for real-world applications, where the actual cost of knowledge acquisition has thus been minimized – keeping the user in the loop. However in web searching, the user is always in the loop, providing additional priorities and relevance feedback as part of the learning process.

4 Results to Date

The web-based search engine to interface the KASER with a user is currently being implemented. After considering several client side and server side technologies, it seems that a server side approach based on a web server browser client model is the best solution. Although several scripting languages are possible to use, PERL so far seems to be the best solution because of its text parsing abilities and existing set of APIs.

The KASER, as an information provider, is also a learning environment. Recent investigation into learning environments and learning dynamics suggests that avatars may be used to create an "engaging, immersive environment". Work in cognitive science has confirmed that receiving information along multiple channels improves understanding and retention. Avatars inject characters into an otherwise stale environment of information-presentation. They, like classroom teachers, aid learners' focus by directing attention to themselves. A good avatar is able to provide a "face" to an application as well as improve information transmission.

The KASER has a Custodian avatar to inject a multimedia experience into the user interface. The avatar, which has been designed, acts as the narrator of the text that is returned by the search engine for the query submitted by the user. The KASER Multimedia System GUI has been modified to support Pedagogical Mode with the addition of a JPanel near the bottom of the window, as can be seen in Figure 2. This JPanel contains a group of radio buttons that allow the user to rate the usefulness of the information returned by the search query. As an example, the user may evaluate text information as (i) useful, if the text is relevant and informative; (ii) not useful, if the text was not relevant or not informative; or (iii) unclear, if the usefulness of the text is unclear. When the user clicks on the Evaluate Text button, the selection will be returned to the KASER, which uses this information to further train the system on which pages it should display in the future.

As can be seen in Figure 2, the text box, which displays information has been expanded and blended into the background of the GUI window to allow for easier reading of the presented text. The GUI window has also been modified to allow the user to resize or maximize the window. As the user changes the size of the GUI window, the textbox and the thumbnail images will grow and shrink to remain in proper proportion and location.

The current plan will have the Custodian avatar displayed in the top center of the window (Figure 3). It will be animated through five or more states: one to represent the Custodian in its dormant state with darker colors; and four or more others to be cycled through while the returned text is being read by the synthesizer and cycling coloration through the suggested inlaid "gemstones".

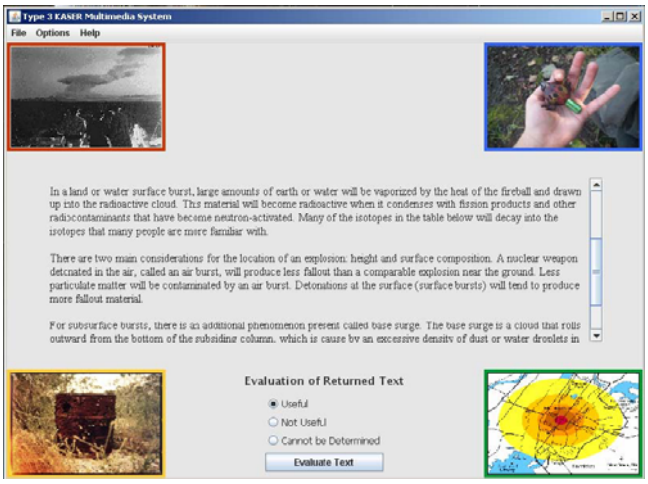

Fig. 2. Initial Implementation of the GUI Web Search Engine

An evaluation selection panel has been added to allow the user to train the model during pedagogical mode. The evaluation selection panel will only appear while the system is in pedagogical mode and will disappear when the system is switched to decision support mode. When the user clicks on the Evaluate Text button, the system will store information about the current text to allow the system to more accurately display text in the future.

The required static IP to all web clients has been implemented to connect and search without special setup (thus bypassing writing a specific client plug-in). Further, a hashing implementation has been extended to uniquely identify and prioritize search results; this will enable cross web search engine capabilities such as allowing Yahoo © and Google © search results to be presented on the same page. Additionally, the required database tables were created to allow for this stateless comparison between pages.

Future work for the avatar includes making the avatar more interactive and thereby forms a stronger connection with the user of the KASER Multimedia System interface. We will also explore methods for being able to more closely manipulate the various phases of the Custodian's animation so as to have a more interactive relationship between the Custodian and the speech synthesizer. One technique that has been implemented in other search applications is the highlighting of keywords in the text. This allows the user to visually scan the page

and find what he is looking for faster and more efficiently. What was implemented in the system is very similar to this. Its function is to select the keywords in the text that closely relate to the keywords found on the web pages the images have been extracted from; highlighting each keyword in a different color, as can be seen in Figure 3. Since the system refreshes the results every time the user scrolls through the page, the highlighted words function will update synchronous with it and will erase any highlights from previous searches.

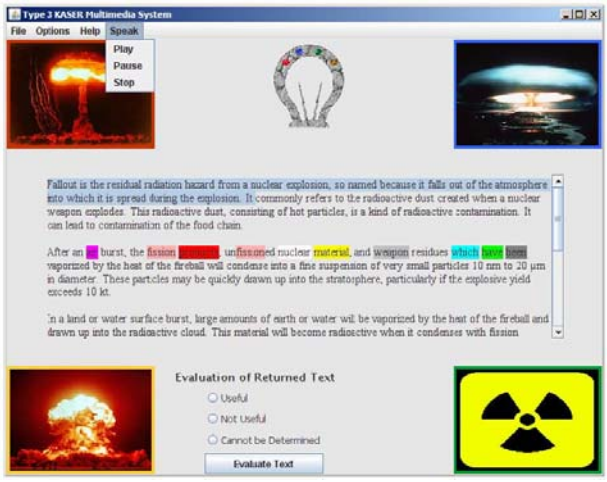

Fig. 3. Version 2 of the GUI Web Search Engine

The KASER's capability to leverage textual databases as well as other types of data such as sensor information provides the user with information where it is needed, when it is needed, and to whom it is needed. It will operate in one of two modes as determined by the user; namely, training mode or decision support mode. These modes automatically assemble instructional material from a very large database of textual information. The user is empowered because he/she does not have to spend anywhere near the time in assimilating otherwise crude records, which are not necessarily directed to their information needs.

5 Concluding Remarks

Currently, searching the web for information, including information generated from sensor networks, is a time-consuming process because the computational system offers the user little if any assistance in processing the more or less relevant websites returned. The potential gains in efficiency are enormous because not only can the system automatically scan and extract salient information, while excluding stated irrelevant information, from say hundreds of thousands of websites (many of which will have redundant information), but the speedup does not adequately reflect the automatic incorporation or exclusion of information that would not be practical for the user to do manually.

The KASER described in this paper is the first step in automatically answering the questions posed by users posed against a free-form text base. The use of positive and negative instances of what to search for supports a computing with words [4]. Therein lies an opportunity to advance the KASER. The further capability required here is to be able to map constraints using analogy. For example, if in searching for information on say lasers, one states that sailboats are a negative instance, one would like the system to understand on the basis of analogical knowledge that motorboats, toy sailboats, etc. are also negative instances. As another example, a commander may be monitoring the battlespace and needs to search the various sensor nodes to acquire information on the status of the battle in different locations. The best way to accomplish this is to employ advanced KASERs, currently under development, that learn such commonsense knowledge from the user and expand it through application.

While such paradigms for computing with words will no doubt advance, the KASER discussed here is the first system of its kind to offer the user capabilities to reduce search time and improve upon the quality of the information returned when searching the web. The gods did not make the heavens overnight. We feel it to be extremely important that such search advancements be disseminated and advanced inasmuch as "knowledge is power".

References

1. Rubin, S. and Lee, G., "Learning Using an Information Fusion Approach", Proc. of the ISCA Int'l Conference on Intelligent and Adaptive Systems, Nice (2004)
2. Rubin, S. and Lee, G. "On the Use of Randomization for System of Systems (SoS) Design of Intelligent Machines", Proc. of the World Automation Congress, ISSCI, Budapest (2006)
3. Chaitin, G.J. , "Randomness and Mathematical Proof," Sci. Amer., vol. 232, no. 5, pp. 47--52 (1975)
4. Zadeh, L.A., "From Computing with Numbers to Computing with Words – From Manipulation of Measurements to Manipulation of Perceptions," IEEE Trans. Circuits and Systems, vol. 45, no. 1, pp. 105—119 (1999)
5. Rubin, S.H., Rush, Jr., R.J., Boerke, J. and Trajkovic, Lj., "On the Role of Informed Search in Veristic Computing," Proc. 2001 IEEE Int. Conf. Syst., Man, Cybern., pp. 2301--2308 (2001)
6. http://www.emma-expertsystem.com/38160S.html

ACKNOWLEDGMENTS
This research was supported, in part by DTRA contract number BA08MSB008. The statements, findings, conclusions, and recommendations are those of the author(s) and do not necessarily reflect the views of the sponsoring agency. This work was produced, in part, by a U.S. government employee as part of his official duties and no copyright subsists therein. It is approved for public release with an unlimited distribution.