A fully automatic question-answering system for intelligent search in e-learning documents.

Kumar, Praveen, Shrikant Kashyap, Ankush Mittal, and Sumit Gupta. "A fully automatic question-answering system for intelligent search in e-learning documents. " International Journal on E-Learning. 4.1 (Jan-March 2005): 149(18).

E-learning is a novel method for presenting information to students for the purpose of education. Currently a sea of information is available in the form of PowerPoint slides, FAQs and e-books. However the potential of this large body of information remains unrealized due to lack of an effective information retrieval system. Current search engines are used only for the web and return ranked lists of documents. Such engines would not be effective searching tools for e-learning documents and it would be difficult for a user to find the intended answer. This article introduces a fully automatic Question-Answering (QA) System that allows students to ask a question in common language and receive an answer quickly and succinctly, with sufficient context to validate the answer. The system uses Natural Language Processing (NLP) techniques to identify the semantic and syntactic structure of the question. It configures itself to a particular domain by automatically recognizing the entities from the course material. The information retrieval engine is used to extract answer passages using contextual information. A closed loop dialogue with the user leads to effective answer extraction through extensive passage analysis. Experimental results of the system are shown over the course material of Computer Networks.

****

E-learning has underlined the importance of quick access to relevant study material for effective education with the major advantage of enabling people to access learning facilities regardless of their location and at the time that is most convenient to them. Business enterprises are widely using this online learning for employee training and education because of its cost saving advantages, especially with respect to time and travel parameters (Dorai, Kermani, & Stewart, 2001). Currently a sea of information is available in the form of PowerPoint slides, digital text and FAQs. However, a lot of time is spent on e-learning by users searching for a desired concept or answer from this huge repository of information.

To fill in this gap, an effective Question-Answering (QA) system is required that can retrieve answers to students' questions from the course material, suggest alternatives in case of any ambiguity in the question and thus help them to search for the intended answer. Examples of such interactive closed-loop QA systems developed are HITIQA (Small, Liu, Shimizu, & Strzalkowski, 2003) and SPIQA (Hori, Hori, Isozaki, Maeda, Katagiri, & Furui, 2003). The rapid success of distance education has led to extensive development of course material and its placement on web. A learner does not understand and knows where he can find the related terms and concepts mentioned in the lecture. Searching for topics through table-of-contents or index pages can be tedious and impractical due to a large volume of information present in these domains. For instance, the user wants to know which algorithms sort an array in a particular time complexity (i.e., O(nlogn)). Since such algorithms are distributed throughout the book (like BinarySearch, Mergesort, Binsort, Radixsort, MinHeap sort, etc.), table-of-contents or index pages cannot provide the user much information and he has to search through the entire book.

Modern search engines (such as Google) are able to cope with the amount of text available. They are most useful when a user presents a query to the search engine which only returns a couple of documents of which the user can then manually search to find the relevant information. Such engines would not be effective searching tools for e-learning documents and it would be difficult for a learner to find the intended answer from the list of retrieved documents. Searching for a particular concept by keyword or phrase matching is insufficient because in many cases (i.e., for the question, "What is the difference between RIP and BGP protocol?") words like "difference" may not be present; instead, words like "compare" or "contrast" can be there. In other cases like, "Give the time complexity of Mergesort," some semantically related terms like "asymptotic" or "Big O notation" have to be identified.

The approach taken here is to implement a QA system based on searching in context and entities of a domain for effective extraction of answers to even domain specific questions. The system recognizes the entities by searching from the course material. It is fully automatic as it does not require any manual intervention for configuring it to any particular domain. The focus is on context based retrieval of information. For this purpose a retrieval engine that works on locality-based similarity heuristics is used to retrieve relevant passages from the collection, (i.e., passages that can potentially answer the question). During query formulation and expansion, the system tries to make judicious interpretation in order to tap the semantics of question. The system utilizes natural-language parsers and heuristics in order to return high-quality answers. This system can be used to serve as a first step towards automatic FAQs. It has good utility for a novice in a subject who does not know where to find related terms and concepts. It can also be quite helpful to students just before their exams for getting answers to review questions.

Contribution of the Article

The following are the contributions of the article:

* Automatic Entity Recognition: The system is not restricted to only one domain. It is fully automatic as it learns about the domain by recognizing the entities from the course material. Manual development of structured data or annotations (as commonly used in other systems) is not required.

* Integration of Alternative Resources: Different e-learning documents like scanned books and PowerPoint slides have different information and presentation methods. Books are illustrative and give detailed analysis of concepts. Slides are condensed, highlighting the key points. Moreover the collection of material may be comprised of books and slides of different authors and teachers (who present the subject in different styles and concepts). The system tries to integrate information from different types of documents and present the summarised answer to the user.

* The system's ability to recognize the context of the problem by using locality based similarity heuristics and query expansion (with the help of WorldNet).

* Closed loop Q & A: The user is provided with a feedback of related keywords which can help the user to reframe a relevant question (within the limits of e-learning materials) and extract the answer from the system.

Organization of the Article

The rest of the article is organized as follows. The Literature Review and Background section gives an account of related work in e-learning and provides background on Question-Answering. The QA System section describes the different components of this QA system in detail. The Results section provides the results of the experiments and the method adopted to test the system's utility. Conclusions and future work follow these sections.

LITERATURE REVIEW AND BACKGROUND

Related work in E-learning

Efforts have been made in the direction of providing ease to the student in extracting information from e-learning documents with respect to effective retrieval and presentation of knowledge. A similar system COVA (on content-based retrieval) enables remote users to access specific parts of interest from a large lecture database by contents (Cha, 2002). However, manual development of XML schemas or annotating the vast amount of information can be laborious and impractical. Another approach introduces Genetic Algorithms into a traditional QA system which uses the concept of Case-Based Reasoning (CBR) (Fu, & Shen, 2004). The huge number of cases that would be generated with large repository (with continual growth) and failure in case of complex queries put limitation in its practical use.

A different approach taken in knowledge-based content navigation in e-learning applications presents a prototype implementation of the framework for semantic browsing of a test collection of RFC documents (Mendes, Martinez & Sacks, 2002). They propose the use of fuzzy clustering algorithms to discover knowledge domains and represent those knowledge domains using TopicMaps. However, success largely depends on how accurately the clusters are identified and the representation still suffers from the drawback attributed to table-of-contents page.

E-learning Media Navigator (ELM-N) from IBM Research is a system with which a user can access and interact with online heterogeneous course materials (Dorai, Kermani, & Stewart, 2001). Their efforts are aimed to reduce human effort and manual annotation work in order to make the system viable for voluminous information. Furthermore, challenges remain in the area of easy-to-use content delivery, access and augmented interaction.

Background on Question Answering

A QA system provides direct answers to user questions by consulting its knowledge base. It attempts to allow the user to ask questions in natural language and receive an answer quickly and succinctly, with sufficient context to validate answer (Hirschman, & Gaizauskas, 2001). Some QA systems that cater to a specific domain have been developed at very early stage. LUNAR (Woods, 1973) was such a closed domain QA system that it only answered questions related to moon rocks and soil gathered by the Apollo 11 mission. However, it relied on having the data to be available in a highly structured form and not as completely unstructured text.

The availability of huge document collections (for example, the web itself), combined with improvements in information retrieval (IR) and Natural Language Processing (NLP) techniques, has attracted the development of a special class of QA systems that answers natural language questions by consulting a repository of documents (Cody, Oren, & Daniel, 2001). Most of the QA systems that have been developed treat the web as a collection of documents and thus cater to huge variety of questions. One of the commercial search engines known as AskJeeves responds to natural language questions, but its recall is very limited because the search engine uses its knowledge base (which is at least partially hand constructed) to answer questions and to update the knowledge base when asked a question which it has not encountered before.

Another QA system, MULDER (Kwok et al., 2001) is claimed to be the first general-purpose, fully-automated question-answering system available on the web. MULDER's architecture, relies on multiple search-engine queries, natural-language parsing, and a novel voting procedure to yield reliable answers (with a recall of the same level as that of Google). However, the difficulty of NLP has limited their ability to give accurate answers to questions that are quite specific to a domain. In addition to the traditional difficulties associated with syntactic analysis, there remains many other problems to be solved, (e.g., semantic interpretation, ambiguity resolution, discourse modelling, inference, common sense, etc.).

QA systems on the web try to answer questions that require a fact or one word answer. This is difficult for questions that are specific to a domain because the targeted domain is unrestricted and no assumption can be judiciously made. E-learning questions are more complex than TREC-type questions as they require domain knowledge and long answers need to be extracted from multiple documents. Moreover these questions have inherent ambiguity. The objective is to allow the user to submit exploratory, analytical, non-factual questions such as, "How does Mergesort sort an array?" The distinguishing property of such questions is that one cannot generally anticipate what might constitute the answer. While certain types of things may be expected, the answer is heavily conditioned by what information is available on the topic. Users generally prefer answers embedded in context, regardless of the perceived reliability of the source documents (Lin, Quan, Sinha, Bakshi, Huynh, Katz, & Karger, 2003). When users search for a topic, an increased amount of text returned significantly decreases the number of queries that they pose to the system.

The QA System

Figure 1 shows the architecture of our QA system. The user begins by configuring the system to the particular course domain by triggering the Automatic Entity Generator module which recognizes domain specific entities from that particular course's documents. The question submitted by the user is classified in Question Classification to identify its case. The question is parsed using the Link Parser which constructs the linkage structure of the question. This information is used for extracting relevant information (like part of speech) during Question Parsing. Subsequently, Query Formulation translates the question into a set of queries that are given as keyword input to the Retrieval Engine. Query Expansion is needed to tap the semantic of the question and improve the answer extraction. The engine returns top passages after weighting and ranking them on basis of locality. Finally, Answer Selection is done by further extensive passage analysis, and is then presented to the user. To improve answers (if the user is not satisfied) the system takes user feedback which is again followed by answer extraction and selection. Each part is described in detail in the next section.

Question Classification

The Question Classifier used pattern matching based on wh-words and simple information to determine question types. The questions were broadly classified into the following categories:

* Questions containing the keywords such as 'various,' 'ways,' 'difference,' 'types,' and 'compare.' These keywords require answers to be extracted from more than one passage. For example, "What are the various algorithms for sorting an array in O (nlogn) time complexity?" or, "What is the difference between RIP and BGP?" Normally, answers to such questions need to be extracted from several passages.

* Questions that ask for numerical data or date. Such questions were identified by a wh-phrase ("How many?", "How tall?", "When?"). The answer passages must focus on numerical data.

* Questions that can be answered from one passage. The Question Focus (object of the verb) is used to find the relevant answer.

Question Parsing

Usually search engines use keywords from the question to construct queries neglecting unimportant words like 'of,' 'for,' 'at,' etc. No importance is given to the syntactic structure of the question while picking up keywords. In such cases the meaning of the question is lost. For example, no difference exists among the questions 'how,' 'why,' or 'what.' This QA system uses Link Grammar Parser to parse the question in order to determine its syntactic structure. This structure is then used to extract part-of-speech information. The Question Focus is identified by finding the object of the verb. Also, Noun Phrases are identified to tap the semantic structure of the question. This information is used to select plausible answers from the e-learning materials.

The Link Parser is a syntactic parser of English, based on link grammar, an original theory of English syntax. Given a sentence, the system assigns to it a syntactic structure, which consists of a set of labelled links connecting pairs of words. (Temperley, Sleator, & Lafferty, 1993). The parser has a dictionary of about 60,000 word forms. It has coverage of a wide variety of syntactic constructions, including many rare and idiomatic ones. The parser is robust; it is able to skip over portions of the sentence that it cannot understand, and assign some structure to the rest of the sentence. It is able to handle unknown vocabulary, and make intelligent guesses from context and spelling about the syntactic categories of unknown words. It has knowledge of capitalization, numerical expressions, and a variety of punctuation symbols.

The Link Parser works as follows. The dictionary of nouns, verbs, adverbs, prepositions and adjectives is used to parse a sentence. The parser starts at the right end and searches linkages throughout the sentence. It considers each entry for the word as a different word and generates all linkages found for all entries. This parser considers relationships between pairs of words. For example, in the sentence shown in Figure 2 there is an S (subject) relation between "Internet" and "is," and a D (determiner) relation between "a" and "network."

The requirements, like parts of speech, syntactic functions and constituents, can be recovered from the link structure rather easily. For example, whatever word is on the left end of an S-link is the subject of a clause (or the head word of the subject phrase); whatever is on the right end is the finite verb; whatever is on the left-end of a D-link is a determiner, etc. The system finds the question focus by using the S or O linkage to get the object of the verb. Importance is given to question focus by assigning it more weightage during retrieval of answers. Moreover, all nouns, verbs, and adjectives in the dictionary are subscripted (as ".n," ".v," or ".a"), so in these cases the syntactic category of the word is made explicit.

The constituent structure of sentences, while not absolutely explicit, is also quite close to the surface in linkage structures. Constituents can be defined as sets of words which can be reached from certain links, tracing in a certain direction. For example, a verb phrase is everything reachable from an S-link, tracing to the right--that is, not tracing through the left end of the S-link itself. For noun phrases there are several possibilities. Anything that can be reached from an O-link by tracing right is an NP (noun phrase). The system tries to find all possible NP in the question. For example, the following NPs were found in the question, "Why are buffers needed at the output port of a switch?"--[buffers], [the output port of a switch], [the output port], [a switch].

Automatic Entity Recognition

This module tries to recognize the entities in a particular course (domain specific entities) to which the user wants to pose questions. This configures the system automatically to any type of course domain. The system administrator on the server providing distance learning (or the user who wants to search answer from documents present in his local system) gives an index or table of contents file as input. The module runs Link Parser on every line giving its syntactic structure. It takes nouns, adjectives and verbs (ending with ing) as entities (as they carry the focus of the sentence). In the absence of table of contents or index pages, the system searches through the main heading and sub headings of slides or digital text for recognizing the entities. If no linkage is formed it tokenizes the string and word filtering is done to remove any elementary words (as shown in Table 1). If no elementary words are found in the string then the whole string is also taken as an entity (for example, Binary Search Tree). The output is stored in the Entity File for subsequent use. This file contains domain specific entities.

Query Formulation

The query formulation module converts the user's question into a set of keywords (query) which is then sent to the retrieval engine for answer extraction. The system uses the entity file to recognize the domain specific entities in the question. During initialization, the system reads from default file (which can be set to a particular course by the user) and constructs a hash table of these entities. Individual words in the question are compared from this table to identify the entities. These keywords are considered most important and are given the maximum weightage of 2.

The question focus (object of the verb) identified during question parsing is also given the same weightage of 2. Elementary words (as shown in Table 1) are given the weightage 0. The rest of the words in the question are given the weightage 1.

Query Expansion: Extending the query through query expansion enhances the search process by including semantically related terms and thus retrieves texts in which the query terms do not specifically appear (Gonzalo, Verdejo, Chugur, & Cigarran, 1998). For example, in questions like, "Compare and contrast link state and distance vector routing algorithm," the answers may occur in sentences such as "The difference between ..." The system uses a popular thesaurus called WordNet to identify semantically related concepts. WordNet is a semantic network containing words grouped into sets called synsets. Synsets are linked to each other by different relations such as synonyms, hypernyms and meronyms. For nouns, the most common and useful relation is the is-a relation. This exists between two concepts when one concept is-a-kind-of another concept. Such a concept is also known as a hypernym. For example, a computer is a hypernym of machine. This creates a network where related concepts can be identified (to some extent) by their relative distance from each other.

Only those query terms were expanded which do not occur as domain entities. Gaining from this knowledge, query evaluation is no longer restrained to query terms submitted by users but may also embody synonymous or semantically related terms. However, caution is taken as these newly found terms are not as reliable as the initial terms obtained from users. Only closely related terms are taken that have direct relation with either the query term itself or with the words that are directly related to the query term. An appropriate weighting (0.5) scheme allows a smooth integration of these related terms by reducing their influence over the query.

Answer Extraction

To extract passages from the collection of documents an Information Retrieval engine is needed to analyse the keywords and passages in detail. The answers to a query are locations in the text where there is local similarity to the query, and similarity is assessed by a mechanism that employs as one of its parameters the distance between words (Kretser, & Moffat, 1999). For this purpose it was found that the locality-based similarity heuristic (in which every word location in each document is scored) provides retrieval effectiveness as good as the document-based technique, and has the additional advantage of presenting focussed answer passages (instead of the whole document) with sufficient context to validate the answer. Therefore, the engine used is based on this concept and has been customized for this application.

The important features of Locality-Based Retrieval (with Similarity) in this context are:

* The focus is on local context by considering top n ranked passages, instead of the top n documents.

* Each term has a certain scope, where its importance decreases with respect to the distance from that term.

* Similarity is computed as the sum of weighted overlaps between terms. It is based on intuitive notion that the distance between terms is indicative of some semantics of the sentence.

The entire retrieval process is carried out using a world-level inverted index using all of the terms in the automatically generated query. An example of a construction of word-level inverted page list is shown in Figure 3. The drawback of the seamless approach is that more index information must be manipulated and that querying requires more resources, but with the use of appropriate techniques these costs are manageable. Using this fully automatic mechanism, results as good as or better than comparable document-based retrieval techniques, and are obtained within relatively modest resource requirements.

Rather than considering the text collection to be a sequence of documents, it is considered to be a sequence of words, and query term occurrences within the collection are presumed to exert an influence over a neigh-bourhood of nearby words. Then, supposing that the influence from separate query terms is additive, the contribution of each occurrence of each query term is summed to arrive at a similarity score for any particular location in any document in the collection. This concept is illustrated in Figure 4.

The contribution function [c.sub.t] is then defined in terms of l, the location of the query term (as an integral word number); x, the word location at which we seek to calculate a contribution; [h.sub.t], the peak height assigned to the term, assumed to occur at the word position occupied by the term in question; and [s.sub.t], the one-sided spread of the term. The parameters that are used for scoring the passages are:

* N: Total number of terms in the collection

* Term frequency ([f.sub.t]): How often the term t appears

* [F.sub.q,t]: Within query frequency of the term

* Inverse document frequency (idf): log (N/[f.sub.t])

* Height ([h.sub.t]): The height assigned to a term t is a monotonic function of the term's scarcity in the collection.

[h.sub.t] = [F.sub.q,t] * log (N/[f.sub.t])

* d = |x - l| is the distance in words between the term in question and the location at which its influence is being evaluated. In each case the value of ct(x; l) is defined to be zero when |x - l| > [s.sub.t]

[C.sub.t](x, l) = [h.sub.t] * [square root of (1 - (d/[s.sub.t])[.sup.2])]

The top N (value set by the user) ranked passages (window surrounding the location) is returned after scoring all the locations of the query term according to the weightage assigned to them.

The implementation also handles case folding and stemming (to match up a keyword with any of its other grammatical forms) of words while searching the words and indexing them into the inverted page list. For repeated use, the system can be configured to reduce the retrieval time manifold. This is done by searching all the domain specific entities (as already identified) from the documents and indexing them into the table beforehand. This increases the speed of the system since each time the question is asked most of the query terms location are already available and the system does not need to search again (except in the case when additional documents have been added).

Answer Selection and Presentation

The top-ranked passages which are now returned (after weighting and ranking on basis of locality and context) are answer candidates. These are further processed to select those answer passages that will be presented to the user. Some passages may be ranked higher just because of frequent occurrence of one of the principal terms in the query without actually illustrating the intended relation for which the user has asked. For example, the user gives following question: "What is the difference between Bus and Star network topologies?" It is probable that a passage from the introduction of network topology (where occurrence of "network" and "topology" is more frequent with just a reference to Bus and Star topology). To avoid these situations the system searches the occurrence of Noun Phrases (identified in the Question Parsing section) in the passages. Those passages in which matches are found are ranked higher amongst the top ones.

After phrase matching, the system processes the passages according to the classification done in question classification. If the question was classified in the second category requiring any date or numerical expression then the system searches for these terms in the passages to match the answer type. For questions in the first category, the system extracts information from more than one passage (those which are scored higher than a threshold value) and presents all of them to the user along with the links to their respective locations in the documents (as shown in Figure 5). This helps the learner to quickly find the relevant information from many documents and to understand the concept. Furthermore, if the top passages are coming from different resources (slides or books of different authors) then they are ranked separately (amongst the same type of resource) and best answer passage from each is presented to the user.

Feedback

Feedback is the one of the important parts of the QA system that distinguishes it from other QA systems being used today. It provides interactivity between the user and the system. When the question is ambiguous, proper feedback can guide the user to improve the query or reformulate the question and get the intended answer. This mechanism prevents the system from failing in case of questions where focus was not clear and proper context was not used. It provides feedback to the user by suggesting extra keywords to be included in the query (as shown in Figure 6). This is done through a closed loop dialogue.

Closed Loop Dialogue

The user inputs a question at the specified place in natural language. After the user has entered the question he observes a sequence of passages as probable answers to his question. With the passages hyperlinks are provided so that user can access the documents concerned. If the user is not satisfied with the answers provided he can opt for improving the query. This is done through domain-specific query expansion. In such cases the system goes for extensive passage analysis where domain specific entities are searched from lower ranked passages. These entities are then suggested as extra keywords to be included in the query. This guides the user on how to improve the query or reformulate the question in such a way that can extract relevant answers from the system. The user can choose any number of entities (amongst the suggested ones) which he thinks can improve his question. He may also opt for reformulating the whole question.

RESULTS

The main goal of our experiments was to determine the efficiency of our system to locate the exact answers or give an indication of having the exact answer just near to the retrieved passages. For experiment purpose, a course on Computer Networks was selected. Text books (scanned) of "Computer Networking: A Top-down Approach Featuring the Internet" by James F. Kurose and Keith W.Ross and "Computer Networks 4th Edition" by Andrew S. Tanenbaum were used along with their PowerPoint slides. The questions used for testing were picked from review questions at the back of the chapters and FAQs available on Internet. Also a separate collection of questions was drawn out by a survey among students with their knowledge of computer networks varying from beginners (not familiar with the subject) to students who performed well in the subject.

The questions covered a wide range of topics on computer networks. They were of varying type, complexity and difficulty. Questions were nonfactual, explanatory and required extracting passages from different places. Three results per query were extracted. The results are shown in Table 2 and Table 3. The time for information retrieval was quite negligible and we aim to make it faster in the near future. The percentages of confidence (on average) the system had that the answer was present in the first, second and third passage was on average 100%, 85%, and 65%. In Table 2, the questions that were answered in these passages are given in the second, third and fourth columns. Under the column Directs, those questions were included which were not answered directly, but gave the indication of the exact answer to be contained in the same document (near the retrieved passages). Those questions which were answered, only after taking feedback from the user, were included in the next column. Those questions which could not be answered by the system were included under the column Failed.

In nearly 11% of the questions, our system failed to get the right answer. Amongst these, nearly half of the questions were not within the purview of the material. The rest of the cases were because the frequency of occurrence of keywords factor failed, giving undue importance to certain keywords. In 7.5% of the questions, the answer improved from failure to exact (because of our query expansion technique). It successfully answered questions like, "What is the difference between source-based tree and centre-based trees in ..." by extracting passages from two different documents. The results are quite pleasing and the importance of feedback is made apparent because it improved the system in the case of failure by giving the right answer.

CONCLUSIONS

In this article, a QA system is proposed which can solve a learner's problems to a great extent with minimal human-computer dialogue. Using the concept of entities the system is fully automated to work in any subject domain with some input from human expertise. The system is based on searching in context and utilizes syntactic and partial semantic information. This achieves good accuracy in results. While additional work is required to enhance the speed and prediction accuracy of the system and to enable it to withstand a very high workload, our initial experiments are promising. The system can handle multiple resources as is frequently available in e-learning domain.

The current implementation utilizes only partial semantic information during answer extraction and selection. It is believed that recall would be much higher if these factors were taken more into consideration. Improvement upon the search facility can be done by storing previous queries and links of their respective answers which were accepted by users in full confidence. Fundamental approach used by (Kutay, & Ho, 2003) for the analysis of students' interaction and learning could be helpful in such a design. Such a facility could be used to help future users and will facilitate group learning, although this will be a burden on the memory of the system. In addition, a learner model similar to building a user model as done by (Davis, Kay, Kummerfeld, Poon, Quigley, Suanders, Yacef, 2003) could be used for enhancing accuracy for repeated use by a learner.

Table 1 Examples of removed words

Words Removed			
By	Is	So	As
To	Otherwise	The	Will
An	In	For	Of
Does	At	Are	Did
Be	Over	We	Our

Table 2 Our questions (mostly Review questions and FAQs)

#Questions	ANSWER 1	ANSWER 2	ANSWER 3	DIRECTS	FEEDBACK	FAILED
150	72	15	3	32	12	16

Table 3 Questions collected from survey

Questions	ANSWER 1	ANSWER 2	ANSWER 3	DIRECTS	FEEDBACK	FAILED
25 (experts)	10	5	2	3	1	4
25 (naives)	14	4	1	2	2	2

Note

Source code for the implementation work can be requested at the following email address: ankumfec@iitr.ernet.in

References

Cody, C. T. K., Oren, E., & Daniel, S. W. (2001). Scaling question answering to the Web. Proceedings of the Tenth International Conference on World Wide Web, 150-161.

Cha, G. (2002). COVA: A System for content-based distance learning. Proceedings International WWW Conference (11), Honolulu, Hawaii, USA.

Davis, J., Kay, J., Kummerfeld, B., Poon, J., Quigley, A., Saunders, G., & Yacef, K. (2003). Using workflow, user modeling and tutoring strategies for just-in-time document delivery. AIEd2003. Workshop on Technologies for Electronic Documents for Supporting Learning (Volume X).

Dorai, C., Kermani, P., & Stewart, A. (2001). ELM-N: E-learning media navigator. International Multimedia Conference Proceedings of the Ninth ACM International Conference on Multimedia, 634-635.

Fu, Y., & Shen, R. (2004). GA based CBR approach in Q & A system. Expert Systems with Applications, 26(2), pp. 167-170(4).

Gonzalo, J., Verdejo, F., Chugur, I., & Cigarran, J. (1998). Indexing with WordNet synsets can improve text retrieval. Proceedings of the COLING/ACL '98 Workshop on Usage of WordNet for NLP.Montreal, Canada, 38-44.

Hirschman, L., & Gaizauskas, R. (2001). Natural language question answering: The view from here. Natural Language Engineering.

Hori, C., Hori, T., Isozaki, H., Maeda, E., Katagiri, S., & Furui, S. (2003). Study on spoken interactive open domain question answering. SSPR2003, pp. 111-113.

Kretser, O.D., & Moffat, A. (1999). Effective document presentation with a locality-based similarity heuristic. Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, San Francisco, pp. 113-120.

Kutay, C., & Ho, P. (2003). Using intelligent agents for the analysis of students interaction and learning. AIEd2003. Workshop on Technologies for Electronic Documents for Supporting Learning, (Volume X).

Lin, J., Quan, D., Sinha, V., Bakshi, K., Huynh, D., Katz, B., & Karger, D.R. (2003). The role of context in question answering systems. Proceedings of the 2003 Conference on Human Factors in Computing Systems.

Mendes, M. E. S., Martinez, E., & Sacks, L. (2002). knowledge-based content navigation in e-learning applications. The London Communication Symposium.

Small, S., Liu, T., Shimizu, N., & Strzalkowski, T. (2003). HITIQA: An interactive question answering system, a preliminary report. Proceedings of ACL '03, Workshop on QA. Sapporo, Japan.

Temperley, D., Sleator, D., & Lafferty, J. (1993) Parsing English with a link grammar. Third Annual Workshop on Parsing Technologies.

Woods, W. (1973). Progress in natural language understanding--An application to lunar geology. AFIPS Conference Proceedings, 42, 441-450.

PRAVEEN KUMAR, SHRIKANT KASHYAP, ANKUSH MITTAL, AND SUMIT GUPTA

Indian Institute of Technology, India

pkmaxuec@iitr.ernet.in

shrikuec@iitr.ernet.in

ankumfec@iitr.ernet.in

sumitfec@iitr.ernet.in

Gale Document Number:A132555482