Answering Questions for an Organization Online

Vladimir A. Kulyukin Kristian J. Hammond Robin D. Burke

Intelligent Information Laboratory,
Department of Computer Science,
University of Chicago
1100 E. 58th St., Chicago, IL 60637
kulyukin@cs.uchicago.edu

Abstract

The World Wide Web continues to challenge organizations to make online access to their expertise convenient for their clients. One means of expertise access that many clients find convenient in everyday life is asking natural language questions of the organization. To support it online, we developed an approach to building organization-embedded question-answering intermediaries, called Information Exchange systems. These systems use their knowledge of the organization's structure to answer the clients' questions and to acquire new expertise from the organization's experts. Our approach uses techniques of hierarchical and predictive indexing, combined term weighting, abstraction-based retrieval, and negative evidence acquisition. We illustrate our approach with the Chicago Information Exchange system, an Information Exchange application embedded in one university's computer science department.

Introduction

The World Wide Web (WWW) continues to challenge organizations to make online access to their expertise convenient for their clients. Standard approaches to expertise access such as query languages, knowledge-intensive natural language understanders, and intelligent browsers may be inappropriate for organizations that must answer questions from diverse groups of clients. A query language is a skill that many clients may not possess nor have the time to acquire. A knowledge-intensive approach demands expensive knowledge engineering and may not scale up to multiple domains. An intelligent browser is inappropriate for clients whose questions are not expressible with the available interface options.

Our approach to online access to an organization's expertise is to employ *organization-embedded question answering*. A question-answering system is a search intermediary between the organization's clients and experts. The intermediary is organization-embedded inasmuch as it knows the organization's units and their areas of expertise. The clients are provided with a natural language interface to the organization. The intermediary answers their questions by retrieving answers to similar questions or by transferring them to relevant experts. As the experts answer the incoming questions, their answers are stored for future use.

The Problem

We implemented our approach in the Chicago Information Exchange system (CIE), an Information Exchange system (IES) embedded in one university's computer science department. The class of problems addressed by CIE is best described through an example:

> *X is a CS undergraduate enrolled in CS115, which uses Scheme. X finds the textbook for the class terse and wants to know if there is another book that he could read, too. How does the CS Department make sure that X gets the answer quickly? How can it see to it that once the answer is available it can be reused?*

Our objective is to develop a technology for building question-answering systems that organizations can easily embed into their information infrastructures. We see this technology as applicable in organizations that receive questions on large sets of topics via the WWW.

An Outline of a Solution

Each IES goes through two stages of deployment: *organization modelling* and *expertise acquisition*. Organization modelling relies on the fact that many organizations are structured as single- or multiple-inheritance hierarchies of units (Solomon 1997; Simon 1976). Since each unit handles a specific area of expertise, the collective expertise of the organization is represented as a hierarchy of topics, each of which corresponds to one such area. Figure 1 gives part of CIE's hierarchy of topics. During the expertise acquisition stage, the organization's experts open *information accounts* under the topics about which they want to answer questions. Opening an information account is a protocol by which an expert becomes known to the system through a brief online interview.

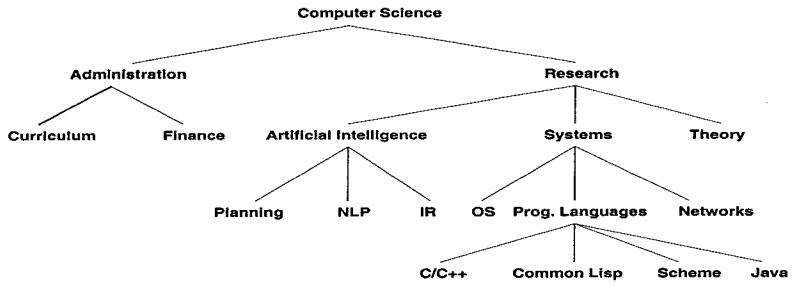

Figure 1: CIE's Hierarchy of Topics

An expert is described by three free-text documents: the description of expertise on the topic from the information account, the collection of question-answer pairs (Q&A's), and the collection of questions classified by the expert as nonrelevant. A topic is described by its subtopics and experts.

The vector space retrieval model is used (Salton and McGill 1983). A collection of documents is a vector space whose dimensions are the documents' terms. Each document is a vector of the weights assigned to its terms. A client's question is a vector in the same space. The document vectors closest to it are retrieved. In addition, several indexing and retrieval techniques are essential to the Information Exchange architecture:

- **Hierarchical Indexing:** The system builds its memory of the organization so as to be searchable by questions. The hierarchy of topics and experts is turned into a hierarchy of vector spaces. Each node in the hierarchy becomes a vector in the space consisting of the vectors of the node's siblings. A topic becomes a vector containing the weights of the terms from its subtopics and experts. An expert becomes a vector containing the weights of the terms from the Q&A collection and the expertise description. Each Q&A becomes a vector containing the weights of the terms from the question, the answer, and the terms found during predictive indexing. Each collection of nonrelevant questions becomes a vector space associated with its expert.
- **Predictive Indexing:** The clients and experts often use different terms to describe the same content. When an expert answers a question, the related terms from a general purpose semantic network are added to the Q&A vector during indexing. If a related term is seen in a question, it triggers the Q&A's retrieval. Information retrieval (IR) researchers proposed several term expansion techniques for similar purposes (Voorhees 1994; Salton and Lesk 1971). Predictive indexing is different insomuch as it is completely automated, goes beyond synonymy and inclusion, and is not based on a single-domain thesaurus.
- **Combined Term Weighting:** A term's weight combines its semantic and statistical properties. Our semantic metric uses the term's part of speech, polysemy, and the closeness of its relation to a term in an expert's question. Our statistical metrics rely on the term's frequency distribution in the collection (Salton and McGill 1983) and its pattern of occurrence in the documents (Bookstein, Klein, and Raita 1998).
- **Abstraction-Based Retrieval:** The system seeks to imitate the behavior of a human search intermediary whose goal is to answer a client's question on the level of abstraction comfortable for the client. The question "How is case-based reasoning used in planning?" is answered on the level of artificial intelligence (AI) by the question "What is case-based reasoning?" whose answer mentions planning in passing. Alternatively, it is answered on the level of planning by the question "What is a case-based planner?" whose answer contains a detailed discussion of case-based planning.
- **Negative Evidence Acquisition:** Relevance feedback is used in IR systems to adjust the term weights in document vectors in response to clients' evaluations (Brauen 1971). But, relevance feedback requires multiple iterations before the term weights improve significantly. Since the experts do not want to receive the same nonrelevant question repeatedly, the experts are indexed by the questions they specify as nonrelevant. When there is a match on a nonrelevant question, the expert is no longer considered. Negative evidence acquistion complements relevance feedback and allows the system to learn from its failures (Hammond 1989).

How CIE Works

CIE is built on four assumptions about indexing and retrieval of free-text expertise. First, expertise on a topic is a collection of Q&A's (Kulyukin, Hammond, and Burke 1996). Second, questions are used as indices into Q&A collections. Third, no answers are generated from scratch: the old answers are reused; the new answers are obtained from the experts when needed. Fourth, the question-answering environment is friendly.

The first three assumptions make CIE a case-based retriever (Kolodner 1993): a Q&A is a case codified for reuse; a question is a cue from the environment that

triggers the retrieval of an answer. The fourth assumption allows CIE to trust feedback from its clients and experts.

Computation of Terms

The terms are computed from the expertise descriptions, Q&A's, and nonrelevant questions. To convert an expertise description, a Q&A's answer, or a nonrelevant question into a term vector, CIE removes from it the terms in its stoplist and applies a greedy morphological analysis to the nonstoplisted words. Our stoplist extends the stoplist derived by Francis and Kucera (1982).

The morphological analysis is based on the morphological component of WordNet, a semantic network of English words and phrases (Miller 1995).[1] The goal is to convert a word to its base form, e.g., the base of "computers" is "computer." The analysis is greedy, because it stops as soon as a conversion rule obtains a base form, which is tagged with the part of speech whose conversion rule obtained it. The parts of speech are nouns, verbs, adjectives, and adverbs. If all rules fail, the word is tagged as a noun.

Computing terms from the question of a Q&A is different. To do the predictive indexing of the question, CIE uses a spreading activation technique (Cohen and Kjeldsen 1987) based on WordNet. WordNet consists of four subnets organized by the four parts of speech. Each subnet has its own relations. For example, nouns have antonymy, the *isa* relation, and three *part-of* relations. WordNet's basic unit is *a synset*, which contains words and phrases interchangeable in a context, e.g., "computer" and "data_processor."

The spreading activation routine uses all four parts of speech, but for each part of speech a subset of relations is used. For nouns, *isa* is used: "machine" is a "computer." For verbs, *entailment* is used: "to limp" entails "to walk." For adjectives, *similarity* is used: "wet" is similar to "watery." No relation is used for adverbs.

The activation routine takes a word and a depth integer specifying how many links away from the origin word the activation is to spread. Each term found during the spread is annotated with its part of speech and the depth at which it was found. Integers 1, 2, 3, and 4 encode nouns, verbs, adjectives, and adverbs, respectively. Thus, "device12" means that "device" is a noun found at depth 2. The origin word's depth is 0. Terms like "device12" are called *annotated* or *a-terms*.

One advantage of predictive indexing is that no activation is spread during retrieval. Instead, the retriever does the *depth expansion* of each a-term in the client's question. For example, if the question contains "machine10" and the depth is 2, "machine11" and "machine12" are added to the question vector, provided that they are dimensions of the current vector space. If an expert's question contains "computer10," and "machine11" is added to the Q&A vector during predictive indexing, the question vector and the Q&A vector match on "machine11."

[1] WordNet is a trademark of Princeton University.

Computation of Term Weights

The semantic weight of an a-term a is computed by the WordNet weight function, $\omega_{wn}(a,r)$, given by $\omega_{wn}(a,r) = \rho(a)/(\pi(a)r^{\delta(a)})$, where $\rho(a)$ assigns an intrinsic weight to each part of speech: 1.0 to nouns, 0.75 to verbs, and 0.5 to adjectives and adverbs; $\pi(a)$ gives a's polysemy; $\delta(a)$ gives a's depth; the rate of decay, r, indicates how much a's weight depreciates with depth. Thus, a's semantic weight is inversely related to its polysemy and its distance from the activation's origin.

The statistical weight combines two metrics. The first metric is known as *tfidf* (Salton and McGill 1983). Let D be the total number of documents in a collection $\mathcal{C}$. Let $\phi(a,\kappa_i)$, $1 \le i \le D$, be a's frequency of occurrence in the i-th document κ_i. Put $\tilde{d}_i = 1$ if $\phi(a,\kappa_i) > 0$, and 0, otherwise. Put $D_a = \sum_{i=1}^{D} \tilde{d}_i$. For a's $tfidf$ weight in κ_i, put $\omega_{tfidf}(a,\kappa_i) = \phi(a,\kappa_i)log(D/D_a)$. The second metric is based on *condensation clustering* (CC) (Bookstein, Klein, and Raita 1998). CC values indexing terms by their patterns of occurrence in a sequence of textual units, e.g., sentences, paragraphs, pages, chapters, and documents. Assuming the sequence proceeds from topic to topic, the terms pertinent to a topic cluster in the units that cover it, while terms that do not bear content appear to be randomly distributed over the units.

Put $\tau(a,\mathcal{C}) = \sum_{i=1}^{D} \phi(a,\kappa_i)$. For the expectation of D_a, put $E(D_a) = DE(\tilde{d}_i)$. Since $\forall(1 \le i \le D)(\tilde{d}_i \in \{0,1\})$, $E(\tilde{d}_i) = 1-(1-1/D)^{\tau(a,\mathcal{C})}$. For a's CC weight in $\mathcal{C}$, put $\omega_{cc}(a,\mathcal{C}) = D_a/E(D_a)$. When a's weight is computed with respect to κ_i, a's local importance is factored in through the product of $\omega_{cc}(a,\mathcal{C})$ and $\phi(a,\kappa_i)$. If a bears content, $\omega_{cc}(a,\mathcal{C})$ is likely to be smaller than 1. Hence, for a's CC weight in κ_i, put $\omega_{tfcc}(a,\kappa_i,\mathcal{C},\theta) = -\phi(a,\kappa_i)log(\omega_{cc}(a,\mathcal{C}))$ if $1 - D_a/E(D_a) > \theta$, and 0, otherwise, where $0 \le \theta < 1$. Let α_1, α_2, and α_3 denote how much importance is given to each metric. The total weight of a in D is given by $\omega_{wn}^{\alpha_1}(\omega_{tfidf}^{\alpha_2} + \omega_{tfcc}^{\alpha_3})$.

Retrieval

Given a client's question, the retriever starts in the top vector space. The client's question Q is turned into a vector of term weights $\vec{Q} = (q_1, ..., q_n)$, where n is the dimension of the current space and each q_i is the weight of an a-term from the question or an a-term added during the depth expansion. The similarity between $\vec{Q}$ and a vector $\vec{V} = (v_1, ..., v_n)$ is the cosine of the angle between them. The similarities between $\vec{Q}$ and the vectors in the current space are thresholded.

If the top retrieved vector is a topic vector, the search proceeds into the vector space under it, which becomes the current vector space. If it is an expert vector, it goes into the expert's nonrelevant collection. If no similarities are found with any nonrelevant questions, the

retriever iterates through the Q&A vectors. If there is a similarity with a nonrelevant question, the expert is no longer considered. Thus, the memory of past failures helps the retriever not to make the same mistake twice.

When several vectors match, the client is asked for his or her search preferences. If the client is unable to determine the relevancy of a topic, the client can examine the description of the topic and the list of experts on it. To determine the relevancy of an expert, the client can see the expert's information account under a specific topic. The client can also search another topic or e-mail his or her question to an expert. Thus, in addition to finding answers to their questions, the clients get insights into the organization's information infrastructure.

The retrieval from an expert's Q&A space is done with the relevance feedback technique similar to the one proposed by Aalbersberg (1992). The Q&A's are retrieved one by one. At each retrieval, the new question vector is formed from the previous question vector and the vector of the previously retrieved Q&A. If the Q&A was relevant, the weights of its terms are slightly increased in the new question vector; if it was nonrelevant, they are slightly decreased. Our technique differs from Aalbersberg's in that the number of negative interactions that the client can have with the retriever is limited. If the client is not satisfied within a certain number of iterations, the client is advised to browse the Q&A collection, contact the expert, or move the search elsewhere. One advantage of this approach is that the retriever knows when to give up.

Upon receiving a client's question, the expert can tell the system that it is nonrelevant, in which case the system adjusts its weights both locally and globally. The local adjustment is made by adding the question vector to the nonrelevant collection. The global adjustment is made by modifying the weights in the vectors on the path from the root down to the expert vector. Since there was a retrieval failure, the idea is to reward the differences between the question vector and the path vectors and to punish their similarities. For the question vector $\vec{Q}$ and each path vector $\vec{H}$ the weights of the common terms are decreased, while the weights of the different ones are increased. This technique is similar to the one proposed by Brauen (1971). However, Brauen's technique takes no action when the term is present in the question vector and absent in a document vector. In our case, these terms are added to the expert's nonrelevant collection.

After the expert answers the client's question, the new Q&A is added to the expert's Q&A collection and then e-mailed to the client. The terms in the Q&A and the terms found during predictive indexing become new dimensions of the expert's Q&A space where the Q&A vector of their weights is added. During the next hierarchical indexing, these terms become dimensions in each vector space that is traversed from the root to the expert vector.

Evaluation

Recall and precision are two evaluation techniques used in many IR systems (Salton and McGill 1983). Recall is the percentage of relevant documents retrieved by a client's question; precision is the percentage of the retrieved documents that are relevant. The information seeking behavior that these measures capture is that of a client who is interested in retrieving all relevant answers.

We observed that the information seeking behavior of the CIE clients is different. A typical CIE client wants to find the first relevant answer fast. One way to measure this is to count the number of interactions the client has with the system before the answer is found. For each test question, we compute the number of interactions that occur as the system searches for the first relevant answer. The number of interactions is averaged over all test questions. Hence, the measure is called the *average number of interactions* (ANI).

We evaluated CIE's ANI with 105 questions about Common Lisp and AI. The questions were obtained from 20 CS undergraduates. Each undergraduate was asked to write 10 questions about Common Lisp and AI that he or she would like to submit to CIE. Out of these 200 questions, a human judge selected the questions that had at least one relevant answer in the system's Q&A collections. In computing term weights, θ was set to .0, α_1 was set to 1.0, α_2 and α_3 were each set to .5.

We first measured the number of interactions it takes a client to find the first relevant answer. Five subjects with a CS background were each given a random sample of size 10 taken without replacement from the test questions. For each question the subjects reported the number of interactions they went through to find the first relevant answer. The following interactions were counted: submitting a question, selecting a topic from multiple topics, requesting a topic's description, selecting an expert from multiple experts, requesting an expert's expertise on a topic, and requesting another Q&A to be retrieved.

The results are summarized in Figure 2. We explain the differences in ANI by the differences in the term ambiguity of different samples. The questions given to the fourth subject had the lowest term ambiguity because most terms identified the correct topic uniquely. For example, in the question "How does garbage collection work in Lisp?", the terms "garbage10" and "lisp10" led to the retrieval of the vectors on the path to the topic of Common Lisp. The questions given to the second subject had the highest term ambiguity, because many terms were indicative of multiple topics. For instance, the terms of the question "Do Lisp process schedulers use the round robin algorithm?" retrieved the topics of Common Lisp, Operating Systems, and Theory. Thus, the second subject had to interact with the system on multiple occasions to clarify her search preferences.

We also experimented with the relative importance of ω_{tfidf} and ω_{tfcc} to see whether the results would

subject	1	2	3	4	5
ani	1.2	3.2	2.5	1.0	3.1

Figure 2: Average Number of Navigation Interactions

Num. of Q&A's/α_2, α_3	20	40	60	80	100	120	140
$\alpha_2 = .7; \alpha_3 = .3$	3.1	2.9	7.5	7.7	8.1	8.6	8.2
$\alpha_2 = .3; \alpha_3 = .7$	6.4	7.2	4.3	4.1	4.7	4.4	4.3

Figure 3: Average Number of Q&A Interactions

confirm our hypothesis that in small collections ω_{tfidf} is a better discriminator than ω_{tfcc}. We chose a Q&A collection of 140 Q&A's about Common Lisp and AI written by a CIE expert. To simulate a dynamically growing collection, we split it into Q&A subsets whose cardinalities were multiples of 20: the first 20 questions, the first 40 questions, etc. For each Q&A subset, we chose the subset of the 105 test questions answered in it. For each question, an ordered list of matches was computed, and the number of nonrelevant Q&A's before the first relevant one was counted. The number of nonrelevant Q&A's was averaged for every Q&A set. The activation depth was 2; α_1 was 1.0. The values of α_2 and α_3 were set to .7 and .3, respectively, in the first experiment, and to .3 and .7, respectively, in the second.

The ANI numbers are given in Figure 3. The table shows that on collections of 20 and 40 Q&A's the metric that valued ω_{tfidf} higher than ω_{tfcc} achieved smaller ANI's, while the metric valuing ω_{tfcc} higher than ω_{tfidf} was more successful on larger collections.

Discussion

How to access online expertise with question answering is an area of active research both in AI and IR (Burke et al. 1997; Kupiec 1993). We share with this research the belief that a good way to make online access to expertise convenient for clients is to allow them to use their language skills. However, our research objective is different. We focus on developing a deployable question-answering technology that organizations can benefit from today.

CIE has much in common with FAQ Finder, a system that provides a natural language interface to the Usenet files of frequently asked questions (Burke et al. 1997). But, while the two systems are similar, because they both combine semantic and statistical techniques in indexing and retrieval, they differ in several respects. CIE predicts indexing features during indexing; FAQ Finder computes them during retrieval. Spreading activation in CIE uses all four parts of speech and several semantic relations; spreading activation in FAQ Finder is restricted to the *isa* relations among nouns. The Information Exchange architecture allows CIE to continuously integrate feedback from the clients and the experts; FAQ Finder does not utilize users' evaluations. CIE treats failures as opportunities to acquire new expertise; FAQ Finder has no explicit notion of failure.

Our approach has its closest precedent in retrievers of case-based reasoning (CBR) systems (Kolodner 1993; Hammond 1989). A CBR system solves new problems by retrieving solutions to similar problems solved in the past. CIE answers new questions by retrieving answers to similar questions answered previously. One unique feature of CIE is that it is a social agent embedded in a real organization (Kulyukin 1998). Its job is to service two user groups with different interests: the clients who are interested in finding fast answers and the experts who want to reduce their question-answering burden. CIE achieves its objective by allowing its memory of the organization to be driven by continuous feedback from its environment.

Conclusion

We presented an approach to building organization-embedded question-answering systems. We outlined the Information Exchange architecture that underlies these systems. One such system is CIE, an Information Exchange application embedded in one university's computer science department. CIE provides clients with online access to the department's expertise by answering their natural language questions. Instead of generating answers from scratch, CIE retrieves the answers to similar questions answered before. When the retrieved answers fail, CIE solicits new answers from the organization's experts. We presented a new metric for computing term weights that uses semantic and statistical properties of terms. We evaluated the metric in several experiments.

Acknowledgments

The authors would like to express their profound gratitude to Dr. Abraham Bookstein of the University of Chicago's Center for Information and Language Studies for his help with several drafts of this paper. They would also like to thank the three anonymous reviewers for their insightful and constructive comments.

References

Aalbersberg, I. J. 1992. Incremental Relevance Feedback. In Proceedings of the 15th Annual International SIGIR Conference, 11-21.

Bookstein, A.; Klein, S. T.; and Raita, T. 1998. Clumping Properties of Content-Bearing Words. *Journal of the American Society for Information Science* 49(2):102-114.

Brauen, T. L. 1971. Document Vector Modification. In G. Salton, editor, *The SMART Retrieval System: Experiments in Automatic Document Processing*, 456-484. Englewood Cliffs, NJ: Prentice-Hall, Inc.

Burke, R. D.; Hammond, K. J.; Kulyukin, V.; Lytinen, S. L.; Tomuro, N.; and Schoenberg, S. 1997. Question Answering from Frequently Asked Question Files: Experiences with the FAQ Finder System. *AI Magazine* 18(2):57-66.

Cohen, P. R., and Kjeldsen, R. 1987. Information Retrieval by Constrained Spreading Activation in Semantic Networks. *Information Processing & Management* 23:255-268.

Francis, W., and Kucera, H. 1982. *Frequency Analysis of English Usage.* New York: Houghton Mufflin.

Hammond, K. J. 1989. *Case-Based Planning: Viewing Planning as a Memory Task.* San Diego, CA: Academic Press, Inc.

Kolodner, J. 1993. *Case-Based Reasoning.* San Mateo, CA: Morgan Kaufmann.

Kulyukin, V. 1998. An Interactive and Collaborative Approach to Answering Questions for an Organization. In Proceedings of the ASIS-98 Mid Year Conference.

Kulyukin, V.; Hammond, K.; and Burke, R. 1996. Automated Analysis of Structured Online Documents. In Proceedings of the AAAI-96 Workshop on Internet-Based Information Systems.

Kupiec, J. 1993. MURAX: A Robust Linguistic Approach for Question Answering Using an On-line Encyclopedia. In Proceedings of the 16th International SIGIR Conference.

Miller, G. A. 1995. WordNet: A Lexical Database for English. *Communications of the ACM* 38(11):39-41.

Salton, G., and Lesk, M. E. 1971. Computer Evaluation of Indexing and Text Processing. In G. Salton, editor, *The SMART Retrieval System: Experiments in Automatic Document Processing*, 143-180. Englewood Cliffs, NJ: Prentice-Hall, Inc.

Salton, G., and McGill, M. 1983. *Introduction to Modern Information Retrieval.* New York: McGraw-Hill.

Simon, H. A. 1976. *Administrative Behavior: A Study of Decision-Making Processes in Administrative Organizations.* 3rd ed. New York: The Free Press.

Solomon, P. 1997. Discovering Information Behavior in Sense Making: III. The Person. *Journal of the American Society for Information Science* 48(12):1127-1138.

Voorhees, E. M. 1994. Query Expansion Using Lexical-Semantic Relations. In Proceedings of ACM SIGIR Conference.