DOrAM: Real Answers to Real Questions

Taras Mahlin
School of Computer Science and Engineering
Safra Campus, Givat Ram
91904 Jerusalem, Israel
taras@cs.huji.ac.il

Claudia V. Goldman
Department of Computer Science
University of Massachusetts
Amherst, MA 01003
clag@cs.umass.edu

Jeffrey S. Rosenschein
School of Computer Science and Engineering
Safra Campus, Givat Ram
91904 Jerusalem, Israel
jeff@cs.huji.ac.il

ABSTRACT

Existing search engines generally retrieve information in response to logical queries consisting of keywords. In contrast, the agent system presented in this paper, DOrAM (**D**omain **Or**iented **A**nswering **M**achine), enables a user to submit a natural language question to the Web, and exploits the question's semantics (along with its keywords) in its search. Moreover, DOrAM improves upon the performance of existing search engines by returning content-based answers to the user's query.

The DOrAM query-response agent is composed of four modules, corresponding to its four-stage approach to question answering. First, the domain of interest is defined in terms of characteristic concepts given by a human expert. Second, documents are gathered from the World Wide Web that are all related to this specialized domain. Third, an ontology is automatically constructed for this domain; this ontology encodes subject-action-object relations found in Web documents gathered in the previous step. The fourth step is the actual question answering stage, returning knowledge that was assembled in light of the user's question and the domain ontology.

Categories and Subject Descriptors

H.3 [**Information Systems**]: Information Storage and Retrieval; H.3.1 [**Information Storage and Retrieval**]: Content Analysis And Indexing; H.3.3 [**Information Storage And Retrieval**]: Information Search And Retrieval

General Terms

Design

1. INTRODUCTION

In this paper, we present a system, DOrAM (**D**omain **Or**iented **A**nswering **M**achine), that combines natural language questions instead of logical queries, and search that leverages specialized knowledge, in our case acquired incrementally, in a specific domain. Moreover, our system attempts to improve upon the typical search engine by retrieving real answers to users' questions. In other words, the user does not receive a list of links through which he needs to browse further in pursuit of the desired information. DOrAM, instead, retrieves an *answer* composed of knowledge acquired during the process of building the relevant ontology.

A key part of our approach is to build an ontology from low-cost representatives of sentences in documents relevant to a certain domain. On the one hand, these sentences do not require deep natural language processing. On the other hand, they still enable us to exploit morphological, semantic, and syntactic constraints imposed by natural language, so that the taxonomy and concept relationships of the chosen domain can be exploited.

AAMAS'02, July 15-19, 2002, Bologna, Italy.

2. THE DESIGN MODEL OF DORAM

Running DOrAM consists of four steps. The domain-based ontology is built through the first three stages. This ontology captures the subject-action-object relations found in relevant documents. The ontology-based graph may further be refined by additional knowledge supplied ("manually") by a human expert. The fourth module actually builds an answer for the user.

2.1 Learning a Domain-Based Ontology

The DOrAM core is based on an ontology built after having acquired morphological, syntactic, and semantic knowledge about a certain domain of interest. First, the domain of interest is specified in a rudimentary way by a human expert. Second, documents found relevant to the basic domain specification are gathered from the WWW. Finally, a graph is constructed after having analyzed representative sentences found in the documents retrieved during the previous stage. This graph should store the information necessary for providing the user with a satisfactory answer.

The Domain Specification: The purpose of this step is to specify a search pattern that will eventually retrieve documents related to the domain of interest. This search pattern is constructed out of the following lists of words: 1. A list of keywords supplied by the human expert. 2. The words found in the submitted sentences together with their syntactic functionality. 3. Additional synonyms, homonyms, and meronyms of each one of the keywords found so far. These additional words are found using WordNet [2].

The search pattern resulting from these three steps is passed on to the next step.

A Specialized Ontology: The search pattern constructed in the previous section is used to gather documents that are all related to a specialized domain. Sentences that are re-

lated to the domain of interest are chosen based on techniques similar to those used in the Musag system [1]. In our implementation, a search tree is constructed out of a root comprising 10 Web pages that were collected by Google[1].

To evaluate the relevance of a page to the domain, the system holds a list of all words encountered so far in the examined pages together with their weights. Once the page that was declared relevant to the target domain is found, the system breaks its content into sentences.

The system thus extracts sentences that complied with the most restrictive rule from the set of heuristics. Unfortunately, we have not yet found a generic set of rules; some rules were good for some pages, and bad for others.

After the system has split the text into sentences, those that are relevant to the domain of interest are chosen for the syntactic analysis.

The Construction of the Ontology Graph: The key idea that shaped the ontology construction stage was to adequately represent sentences in a simple manner, taking into account their morphology. The ontology captures subject-action-object triplets, where the action connects between the subject and the object, and the object can describe the circumstances in which the action took place. Our assumption is that, for a given domain, there will be enough sentences that demonstrate this structure. This structure captures relationships among concepts in the domain and their taxonomy.

2.2 The User-Ontology Interface

To find correlations between the concepts in the user's question and the concepts in the ontology, the user's question is analyzed by 1) extracting its syntactic structure (NP;VP;NP), 2) extracting its question words (how, what, ...), and 3) extracting its keywords.

Several metrics (e.g. concept distance, concept complementation) are defined for measuring the closeness between any two concepts in the ontology. Then, for each keyword in the user's question, DOrAM computes these measures with respect to a given ontology. In this way, the system can answer the user's question with the fittest information found in the ontology.

Next, we classify the question with regard to its distinctive features such as question words or asking points as in Textract [4], concept of interest (as in the Lasso system [3]), and keywords. The answer could be described as a binary answer (yes/no), as a plan answer, or as a general descriptive answer.

The system produces as its final output a subset of its domain-based ontology graph. This output is built in different ways depending on the classification of the user's question.

3. EXPERIMENTS AND RESULTS

We tested three domains of increasing complexity.

The initial dictionary of domain concepts and examples from the additional concepts learned at the end of the information gathering process are shown in Table 1.

In our work we put an emphasis on the open class question characterized by How, What and Why question words. Some of the questions we asked the system upon completion of the preliminary ontology-building phase, and the answers that the system supplied, appear in Table 2.

As we can see from Table 2, the results of the system are generally encouraging, though far from perfect (and still requiring human interpretation). For example, to buy a house according to DOrAM you should get to a city in which you will find tools; these tools will help you to find a house. In addition, you will turn to an agent who also helps you to buy a house.

[1] www.google.com

Sample Domain	Initial dictionary	Learned concepts
Buying real estate	buy ,house agent ,mortgage	loan , paint real , home ...
Travel and recreation	sport , everybody travel , vacation,	skiing , ecotourism fishing , park ...
Art and Painting	culture , color museum, display, art	work , artist collection, web ...

Table 1: Dictionary contents example

Sample Domain	Question	Answer
Buying real estate	What is necessary for buying a house ?	You get cities Cities have real Real help find tools Tools find agent Agent help buying
Travel and recreation	How can I plan a travel ?	Hotels offer service Service is limited time Time will arrange island Island services vacation
Art and Museums	What is a connection between a museum and an artist ?	Museums don't have site Site include paintings Paintings indian media Media are used arts Art maintains artists

Table 2: Question and answer examples

4. CONCLUSIONS AND FUTURE WORK

We have introduced a method for extending and improving the performance of standard information retrieval tools. DOrAM, the system presented in this paper, enables a user to submit real questions to the Web. The system then presents the user with two kinds of output based on a domain-based ontology previously learned. The user gets a list of weighted concepts related to the domain of interest that is also related to his question, and a plan of action that achieves the user's goal (as described in his question).

5. REFERENCES

[1] C. Goldman, A. Langer, and J. Rosenschein. Musag - an agent that learns what you mean. *Journal of Applied AI* 11(5):413–435, 1997. Edited by N. Jennings and B. Crabtree.

[2] G. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller. Introduction to WordNet: An on-line lexical database. *International Journal of Lexicography*, 3(4):235–244, 1991.

[3] D. Moldovan, S. Harabagiu, M. Pasca, R. Mihalcea, R. Goodrum, R. Girju, and V. Rus. Lasso: A tool for surfing the answer net. *In Procs. of TREC-8*, Oct 1999.

[4] R. Srihari and W. Li. Information extraction supported question answering. *In Procs. of TREC-8*, Oct 1999.