Knowledge Acquisition for Question Answering

Roxana Gîrju and Dan Moldovan
Department of Computer Science and Engineering
Southern Methodist University
Dallas, Texas, 75275-0122
{roxana, moldovan}@engr.smu.edu

Abstract

Questions can be classified based on their degree of difficulty. As the level of difficulty increases, question answering systems need to rely on richer semantic ontologies and larger knowledge bases. This paper is concerned with questions whose answers are spread across several documents and thus, require answer fusion. To find such answers, the system needs to develop domain specific ontologies. A method is presented for on-line acquisition of ontological information from the document collection.

Introduction

Question Answering (QA) has attracted considerable interest in the last few years. The explosion of information and the need for new tools that reduce the amount of text to be read in order to obtain the desired information motivate the growing interest in QA systems. As defined by the TREC-QA (Vorhees and Tice, 1999), a question answering system has to identify the answer of a question in large collections of documents by highlighting a small part of text which contains the answer.

The question expressed in natural language is first analyzed and classified based on its type (*who, what, where, when, why*, etc.), and then the keywords are extracted. Questions of low difficulty levels, like *What is the largest city in Germany?*, and *How did Socrates die?* can be answered using predefined semantic dictionaries, and simple NLP techniques that help locate the answer in the collection of documents based on keywords matching and proximity.

The literature shows that for this type of questions, question answering systems can achieve high performance levels based on simple retrieval and highlighting of paragraphs (TREC-8 and TREC-9).

As the level of difficulty increases, question processing needs richer semantic resources, like ontologies and larger knowledge bases. Consider the question: *What are the software products that Microsoft sells?*. For questions like this, the system must first find out what constitutes *software products*, and then check whether or not Microsoft sells such products.

Unless an ontology of software products exists in the knowledge base, which is highly unlikely, the system must first acquire from the document collection what software products are. This on-line ontology development is the problem addressed in this paper.

The state of the art in question answering systems shows that this class of questions has not been addressed yet. The most performant question answering systems today, can extract single facts from a large collection of documents, but are unable to answer questions that require answer fusion.

We believe that dynamic ontologies built ad-hoc from the text collection, coupled with existent ontologies are the best path to follow in answering more and more difficult questions. In order to address questions of higher degree of difficulty, we need to handle real-time knowledge acquisition and classification for different domains.

This paper presents QAAF (Question Answering for Answer Fusion), a module that extends an existent question answering system by handling questions whose answers are scattered across several documents. The answers extracted are organized into a dynamic taxonomy built ad-hoc from the text collection. In the following sections the architecture of QAAF is described and the algorithm used for ontology development is explained. Finally, results are presented and examples of questions solved with this approach are given, as well as comments and future work.

Overview of the System

The QAAF module is an extension of an existing question answering system (Harabagiu et. al, 2000), and its block diagram is shown in Figure 1. The components of the QAAF module are: *question processing*,

sentence indexing, answer extraction, answer classification, ontology development, and *query formulation,* and are described in more detail in the next sections.

The QAAF Question Processing

A question expressed in natural language is processed by the question answering system and three types of information are extracted: *question type*, the *expected answer type* from the semantic analysis of the question, and the *question focus* defined as the main information required by that question. The question processing also identifies the question keywords. For example, for the question *What are the software products that Microsoft sells?*, the focus is the same as the answer type, *software products*, and the keywords are *software products*, *Microsoft*, and *sell*. For questions like *What causes hypertension?*, the focus is represented by a verb, *cause*. This indicates that the answer type should be the cause of an event or state, in this case *hypertension*. For this type of questions, QAAF does not include the focus in the list of keywords.

For each keyword, QAAF extends the question information by extracting related concepts based on an existing ontology and/or other on-line dictionaries. For this experiment we used WordNet(Miller, 1995). The related concepts of the keywords are determined as: synonyms for nouns and verbs (for example, in WordNet *software product* is synonym with *software package*), troponyms for verbs (the troponym of verb *sell* is *market*), and for adjectives, the noun it comes from (if applicable).

Furthermore, based on the focus, QAAF classifies and answers the following categories of ambiguous questions that require answer fusion:

- Definition questions whose focus is an NP.
  For example, *What software products does Microsoft sell?*
- Cause/Effect questions.
  E.g., *What causes hypertension?*

The QAAF Sentence Indexing

With information from the *question processing* phase queries are formed and passed to the *Indexing* module. Depending on the type of question, the query seeds may contain:

- the focus, for definition questions,
- the cause, for effect questions,
- the effect, for cause questions.

The question answering system extracts the documents considered relevant, and additionally it retains only those paragraphs containing this information. Furthermore, because our approach works at the sentence level, QAAF retains only those sentences that contain the seeds.

The QAAF Answer Extraction

The *answer extraction* module identifies and extracts partial answers from the sentences determined in the *indexing* phase. The extraction of the answer is done by the QAAF based on the semantic relations and lexico-syntactic patterns applied on the sentences extracted by the previous module. The sentences are first part-of-speech tagged and parsed by the question answering system using lexical and semantic information in order to identify name entities.

Relations/Patterns Selection

Based on the question type determined in the *question processing* phase, the *Relation Selector* module selects the corresponding relation(s). The relations considered in this paper are: IS-A, PART-OF and CAUSE. For each relation, QAAF picks up from a table the corresponding surface lexico-syntactic patterns through which each relation is expressed. The relations and the corresponding patterns used for this experiment are shown in Table 1.

Relation	Lexico-Syntactic Pattern
IS-A	NP1 such as NP2 NP1, including NP2 NP2..NPn, and other NP1 NP1, especially NP2..NPn
CAUSE	NP1 cause NP2 NP2 caused by NP1 NP1 effect on NP2
PART-OF	NP2 part of NP1

Table 1: Some semantic relations and lexico-syntactic patterns used to build the dynamic ontology from text. The patterns were presented in (Hearst, 1999) and (Moldovan, 2000).

Partial answer extraction for definition questions whose focus is an NP

THE ANSWER EXTRACTION ALGORITHM:

Input: query seed and related concepts, relations, and lexico-syntactic patterns

Output: NPs representing partial answers

1. Determine from the text collection all the NPs that have the seed as head.
2. Select all the patterns corresponding to the IS-A and PART-OF relations:
   - *<NP* IS-A *focus_concept>*
   - *<NP* PART-OF *focus_concept>*,

   where *focus_concept* is the query seed, or any NP determined at step 1. For example, for the sentence *What software products does Microsoft sell?*, one such pattern will be *<NP* IS-A *software product>*.
3. Extract all the IS-A and PART-OF relationships determined above and select all the NPs occupying the hyponym position in these relationships. For example, for the question above, the system finds that *Microsoft Office* IS-A *software package.*

This way, a list of partial potential answers is formed.

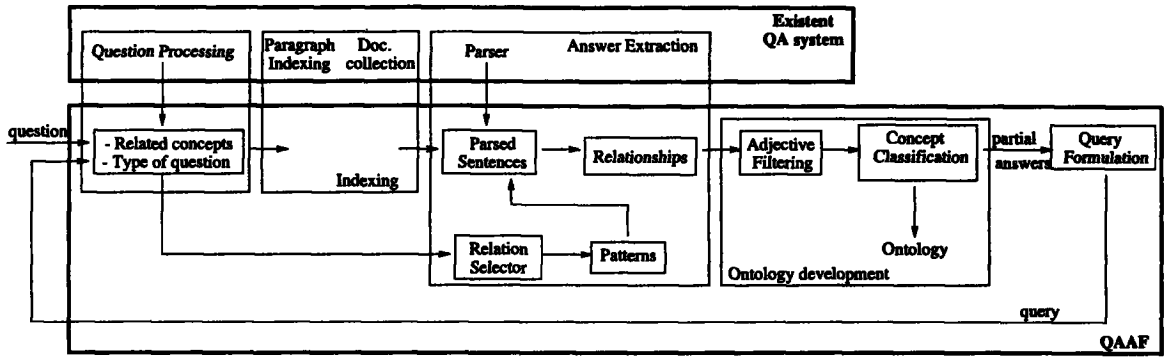

Figure 1: The block diagram of the QAAF module for fusion questions.

Cause/Effect questions

1. Effect questions

These are questions asking for the effects of events or states. For example, *What are the effects of stress?*

Partial Answer Extraction:

Input: query seed and related concepts, relations, lexico-syntactic patterns

Output: NPs representing partial answers

1. Apply on the parsed sentences all the CAUSE patterns <seed_NP/seed-related_NP CAUSE-PATTERN NP>. For the example given above, the pattern will be *<stress/tension* CAUSE *NP>*.
2. Select all the NPs which occupy the effect position in the relationships determined above. For example, the system finds from the text collection that *stress* CAUSE *depression.*

This way, a list of partial answers is formed.

2. Cause questions

These are questions asking for the causes of events or states. For example, *What causes hypertension?*

PARTIAL ANSWER EXTRACTION:

Input: query seed and related concepts, relations, lexico-syntactic patterns

Output: NPs representing partial answers

1. Apply on the parsed sentences all the CAUSE patterns:

<NP CAUSE-PATTERN seed_NP/seed-related_NP>. For the example given above, the pattern will be *<NP* CAUSE *hypertension/high blood pressure>*.

2. Select all the NPs which occupy the cause position in the relationships determined above. For example, the system has determined from the text collection that *obesity* CAUSE *hypertension.*

Ontology Development

Adjective Filtering

After partial answers are extracted, the QAAF has to filter out unimportant modifiers and to construct an ontology. Most of the time, partial answers noun phrases contain adjectives.

As described in (Katherine Miller, 1998), adjectives are divided in WordNet into three categories: *descriptive, participial* and *relational.* The descriptive adjectives constitute the largest category and express an attribute of the modified noun, as in *heavy package*, or *high speed.* These adjectives are organized in clusters defining a particular attribute and can be gradable. For example, the cluster *small, large, tiny, etc* defines the attribute *size*, and *fast/slow* the attribute *speed.*

Participial adjectives are derived from the participle form of verbs and consist of forms ending in *-ing* and *-ed.* For example, *boiling water*, or *married couples.*

The relational adjectives, the second largest adjective class, are adjectives related semantically and morphologically to nouns. Such examples are *musical instrument* or *English muffin.*

Heuristic: Keep as new concepts only those that are formed with relational and participial adjectives and discard those that have descriptive adjectives.

The rational for this is that descriptive adjectives bring more information about the concept on hand, without conferring new, well defined meanings to the nouns they modify. Here are some examples of noun phrases containing descriptive adjectives: *popular software package, important physical problems*, and others. It is our opinion that it is not useful to consider these as new concepts. On the other hand, relational and to some extend participial adjectives are more likely to form new concepts when attached to nouns.

Based on this, the system takes out all the adjectives from the NPs, with the following exceptions:

1. when the adjective is part of a concept determined from WordNet or an on-line dictionary (e.g.: *high blood pressure*), or
2. when the adjective is a relational or participial adjective, such as *operating systems.*

Concept Classification

The classification procedure organizes into an ontology the information determined in the *answer extraction* phase. For the definition questions, the ontology is built under the focus, using IS-A and PART-OF relations. In the case of the cause/effect questions, the ontology is built horizontally, with CAUSE relations. For this type of questions, the starting node is the keyword given in the question. For example, in the question *What causes hypertension*, the starting node is *hypertension*. All the other nodes represent NPs that cause it. Regardless the type of question, the ontology is built incrementally, one level at a time, until no more relationships are found in the text collection. This way, at each iteration, the relationships found in the *answer extraction* phase, are added to the corresponding ontology. In addition to this, some of the noun phrases determined for the definition questions in the *answer extraction* phase, need to be further classified.

THE CLASSIFICATION PROCEDURE FOR THE DEFINITION QUESTIONS WITH A NOUN PHRASE FOCUS:

1. Select all the NPs having as head the focus. Classify them relative to their most specific subsumers, as defined by the *subsumption principle*.[1]
For example, *PowerPoint business presentation software product* IS-A *business software product*, which IS-A *software product*. These NPs were detected in the *answer extraction*.
2. Classify all the remaining NPs extracted by the *answer extraction* module above.

a. For IS-A relationships:

i.) If the hypernym is the focus, then classify the NP under the focus. For example, *software products, including Microsoft Office* $\Longrightarrow$ *Microsoft Office* IS-A *software product*.

ii.) If in the hypernym position there is an NP that has the focus as head, then classify the new NP under the focus: *operating systems software product, such as OS/2.* $\Longrightarrow$ *OS/2* IS-A *operating systems software product* which IS-A *software product*.

iii.) Classify all the remaining NPs that can be classified with the *subsumption principle* to the already classified noun phrases.

b. For PART-OF relationships:

i.) If the focus is in the *whole* position, or the *whole* position contains an NP having the focus as head, then add the relationship to the new ontology as described for IS-A relationships.

ii.) Classify all the remaining NPs that can be classified with the *subsumption principle* to the already classified noun phrases.

[1]This classification procedure is based on the *subsumption* procedures described in (Woods, 1991) and (Moldovan, 2000).

Query Formation

At the end of the *classification* phase, there remains a list of noun phrases that couldn't be classified yet, as no relevant information was found in the sentences retrieved. For each such NP, queries are formed and sent to the *question processing* module where QAAF extends the query. For each query, the system is started again. The reason for this approach is that we need to get all the information that exists in text about each NP. This way, chains of concepts starting with the root node are formed.

In the case of definition questions like *What software products does Microsoft sell?* the ontology built for the *software products* contains at this point all the software packages that could be found with this approach in the text collection. In order to answer the question, the next step is to select from all these software products only the ones that are sold by Microsoft. This procedure is done by the *answer extraction* module of the question answering system based on keywords matching and proximity. Figure 2 shows the dynamic ontology created from the text collection for this question.

Results

The system was tested on 20 questions, ten definition questions, and ten cause/effect questions. On average, the ontologies contain 25 nodes arranged in about 3 levels. This shows that they are bushier than deeper. For the cause/effect questions, the answers were also direct and indirect causes/effects of the root node, and they were represented by all the nodes in the ontology, with the exception of the root. For the definition questions, the answer can occur at any level, and not only as leaves. Another aspect that makes this approach interesting is that the relations and the patterns considered are very frequently used, making it possible to build quite large ontologies from large text collections.

The approach taken in this paper for answering definition questions whose question focus is an NP is a top-down one. This could determine quite a large ontology. Taken the example question *What software products does Microsoft sell?*, for each node the procedure selects initially all the software products that exist in the text collection, without specifying which of them belong to Microsoft. The filtering phase is done at the end when all the ontology is built, by highlighting all the Microsoft software products.

This approach is much more time consuming, but it has the advantage of building an ontology that can be used later, for similar kinds of questions.

Another approach for this example question would be to start with the products that Microsoft produces and sells, and tests if they are software products. At

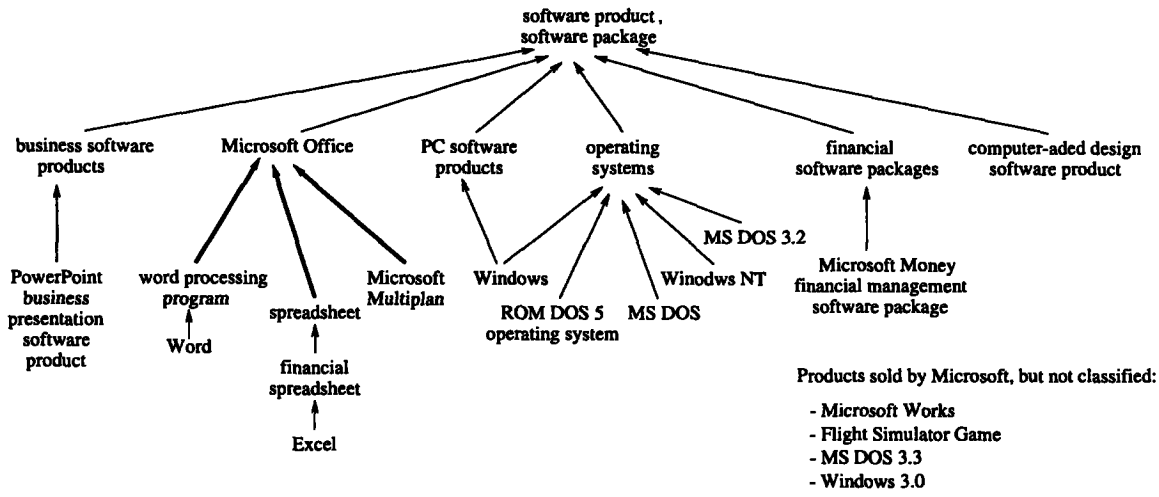

Figure 2: The ontology created for the Microsoft's software products for the question *What software products does Microsoft sell?*. The thicker arrows represent PART-OF relations. All other lines are IS-A relations.

each iteration of the recursive algorithm, it filters out the products that cannot be part of the ontology and the ones for which it doesn't have enough information to classify. This bottom-up approach has the advantage of building a dynamic ad-hoc ontology selecting only the software packages manufactured by Microsoft. This way, it uses only the information related to the question.

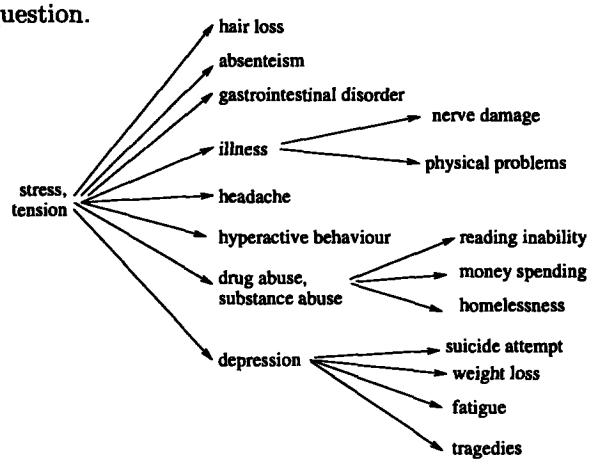

Figure 3: The horizontal ontology created for the effect question *What are the effects of stress?* The arrows represent CAUSE relations.

Figures 2 and 3 show the dynamic ontologies created from the text collection for the two types of questions considered in this paper.

Comments

We believe that the method used in this paper for ontology construction is useful not only for answering fusion questions, but it also represents a good way of building frameworks for reasoning techniques. This way, questions of higher difficulty levels that need world knowledge can be addressed based on very large ontologies.

One of the drawbacks of our approach is that it does not take into consideration complex linguistic phenomena like coreference resolution and word sense disambiguation. Without a good handling of these problems the results are not always accurate.

We also plan to acquire more semantic relations and lexico-syntactic patterns, and to use more advanced NLP techniques, that help in the answer acquisition.

References

Ellen M. Vorhees and Dawn M. Tice. The TREC-8 Question Answering Track Evaluation. In the *Proceedings of TREC-8*, 1999.

Marti Hearst. Automated Discovery of WordNet Relations. In the *WordNet: An Electronic Lexical Database and Some of its Applications*, editor Fellbaum C., MIT Press, Cambridge, MA, 1998.

Katherine Miller. Modifiers in WordNet. In *WordNet: An Electronic Lexical Database*, MIT Press, Cambridge, MA, 1998.

G. A. Miller. WordNet: A Lexical Database. In the *Communication of The ACM*, vol. 38: No.11, November, 1995.

D. Moldovan, R. Girju and V. Rus. Domain-Specific Knowledge Acquisition from Text. In the *Applied Natural Language Processing (ANLP-2000)* conference, Seattle, WA, April-May 2000

S. Harabagiu, D. Moldovan, M. Pasca, R. Mihalcea, M. Surdeanu, R. Bunescu, R. Girju, V. Rus, and P. Morarescu. FALCON: Boosting Knowledge for Answer Engines. In the *Proceedings of Text REtrieval Conference (TREC-9)*, 2000.

W.A. Woods. Understanding Subsumption and Taxonomy: A Framework for Progress. In the *Principles of Semantic Networks: Explorations in the Representation of Knowledge*, Morgan Kaufmann, San Mateo, CA, 1991.