Special Section on Restricted-Domain Question Answering

Question Answering in Restricted Domains: An Overview

Diego Mollá*
Macquarie University, Australia

José Luis Vicedo†
University of Alicante, Spain

Automated question answering has been a topic of research and development since the earliest AI applications. Computing power has increased since the first such systems were developed, and the general methodology has changed from the use of hand-encoded knowledge bases about simple domains to the use of text collections as the main knowledge source over more complex domains. Still, many research issues remain. The focus of this article is on the use of restricted domains for automated question answering. The article contains a historical perspective on question answering over restricted domains and an overview of the current methods and applications used in restricted domains. A main characteristic of question answering in restricted domains is the integration of domain-specific information that is either developed for question answering or that has been developed for other purposes. We explore the main methods developed to leverage this domain-specific information.

1. Introduction

There has been an interest in representing knowledge and automatically processing it from the time of the first generation of computers. This interest has increased from the end of the 1980s to become an urgent necessity. Decisive factors in this increase of interest are an unprecedented growth in the amount of digital information available, an explosion of growth in the use of computers for communications, and the increasing number of users that have access to all this information.

These circumstances have fostered research into information systems that can facilitate the localization, retrieval, and manipulation of these enormous quantities of data. Question Answering (QA) is one of these research fields.

In this article, QA is defined as the task whereby an automated machine (such as a computer) answers arbitrary questions formulated in natural language. QA systems are especially useful in situations in which a user needs to know a very specific piece of information and does not have the time—or just does not want—to read all the available documentation related to the search topic in order to solve the problem at hand.

* Division of Information and Communication Sciences, Macquarie University, New South Wales 2109, Australia. E-mail: diego@ics.mq.edu.au.
† Departamento de Lenguajes y Sistemas Informáticos, Universidad de Alicante, Campus de San Vicente del Raspeig, Apdo. 99. Alicante, Spain. E-mail: vicedo@dlsi.ua.es.

Submission received: 2 June 2006; revised submission received: 15 October 2006; accepted for publication: 23 October 2006.

Research in QA has been developed from two different scientific perspectives, artificial intelligence (AI) and information retrieval (IR).

Work in QA since the early stages of AI has led to systems that respond to questions using the knowledge encoded in databases as an information source. Obviously, these systems can only provide answers concerning the information previously encoded in the database. The benefit of this approach is that having a conceptual model of the application domain represented in the database structure allows the use of advanced techniques such as theorem proving and deep reasoning in order to address complex information needs.

Currently we are witnessing a surge of activity in the area from the perspective of IR, initiated by the Question Answering track of TREC[1] in 1999 (Voorhees 2001). Since then, increasingly powerful systems have participated in TREC and other evaluation fora such as CLEF[2] (Vallin et al. 2005) and NTCIR[3] (Kando 2005). From this perspective, question answering focuses on finding text excerpts that contain the answer within large collections of documents. The tasks set in these conferences have molded a specific kind of question answering that is easy to evaluate and that focuses on the use of fast and shallow methods that are generally independent of the application domain. In other words, current research focuses on text-based, open-domain question answering.

Both trends have developed in parallel and represent the opposite ends of a spectrum connecting what we might label as **structured knowledge-based** and **free text-based** question answering. Whereas structured knowledge-based QA systems are well adapted to applications managing complex queries in a very structured information environment, the kind of research developed in TREC, CLEF, and NTCIR is probably better suited to broad-purpose generic applications dealing with simple factual questions such as World Wide Web–based question answering.

However, both approaches have serious disadvantages when they attempt to tackle important real applications that handle complex questions by combining domain-specific information typically expressed in different sources (structured, semistructured, unstructured, etc.) using reasoning techniques. Examples of such applications are:

Interfaces to machine-readable technical manuals: Many software applications are very complex and they are accompanied by extensive documentation. A QA system that finds specific answers to a user's question based on such documentation would be very useful.

Front-ends to knowledge sources: Many disciplines and areas of human activity have their own specific knowledge sources. An example is the medical domain, which, as we shall see in this article, contains a wealth of technical information and resources that can be used for a QA system targeting this kind of information.

Help desk systems in large organizations: Help desk staff in large organizations need to quickly satisfy the customer's need for information. Although many such requests for information will be found in FAQs available to the help desk staff, there will always be requests that are unique and that require staff to have access to fast methods to find the relevant information. End systems tailored to such staff (who can be trained) are different from QA systems designed for the end user, but they still need to leverage the organization domain.

1 Text REtrieval Conference (http://trec.nist.gov/).
2 Cross Language Evaluation Forum (http://clef.iei.pi.cnr.it/).
3 NII-NACSIS Test Collection for IR Systems (http://research.nii.ac.jp/ntcir/).

It might be argued that focusing research on restricted domains is limiting because the results are too specific and not open to generalization. This may have been the case with early work in natural language processing (NLP), which focused on restricted domains simply because of limitations in computing power and in theoretical coverage. This is not the case nowadays. The availability of comprehensive and reliable resources in complex domains enables interesting and fruitful research to be carried out in restricted-domain natural language processing.

In short, research in restricted-domain question answering (RDQA) addresses problems related to the incorporation of domain-specific information into current state-of-the-art QA technology with the hope of achieving deep reasoning capabilities and reliable accuracy performance in real world applications. In fact, as a not-too-long-term vision, we are convinced that research in restricted domains will drive the convergence between structured knowledge-based and free text-based question answering.

In this article we survey past and current work on question answering in restricted domains. In the process we will highlight the advantages of developing systems based on restricted domains. Section 2 provides a historical note on question answering, with an emphasis on restricted domains, and focusing mainly on early work. Section 3 presents desirable characteristics of restricted domains for the development of NLP research in general, and question answering in particular. Section 4 comments on some of the main factors that distinguish question answering in an open domain from question answering in a restricted domain. Section 5 focuses on the use of domain-specific resources for question answering. Section 6 outlines current restricted-domain question answering methods. Section 7 notes the main aspects to consider when building a restricted-domain question answering system. Section 8 introduces the articles in this special section of the journal, and finally Section 9 presents some conclusions.

2. Early Work

Two examples of early question-answering systems are BASEBALL and LUNAR. BASEBALL answered questions about baseball games played in the American league over one season (Green et al. 1961), and LUNAR answered questions about the analysis of rock samples from the Apollo moon missions (Woods 1997). Both systems were very successful in their chosen domains. In particular, LUNAR was demonstrated at a lunar science convention in 1971, where it was able to answer 90% of questions posed by geologists without prior instructions with regard to the allowable phrasing (Hirschman and Gaizauskas 2001). Both LUNAR and BASEBALL are examples of what have been described as **natural language interfaces to databases**, that is, their source of information was a database that contained the relevant information about the topic. The user's question was converted into a database query, and the database output was given as the answer. The very specific nature of the domains enabled the construction of appropriately comprehensive databases, and a domain-specific question analysis that enabled a mapping from the meaning of the user's question onto the corresponding database query.

LUNAR and BASEBALL are only two of the most salient examples of early work on question answering, but there were many other such systems. Simmons's (1965) survey described a variety of early QA systems. Most of these focused on restricted domains by developing a database of knowledge and providing a natural language interface. Still, many of these early systems (including LUNAR and BASEBALL) were no more than "toy systems" that focused on very limited domains. Those early systems that used a corpus of text as the inherent information source typically processed small volumes of text and would rely on a human to disambiguate the corpus sentences or convert them to a simplified version of English.

During the 1970s and 1980s there was intensive research on the development of theoretical bases for computational linguistics. This research prompted the development of QA systems on domains that were more complex than those of the earlier systems. The main goal of this research was to use QA as an application framework within which general NLP theories could be tested. This work culminated in large and ambitious projects such as the Berkeley Unix Consultant (Wilensky et al. 1994).

The Berkeley Unix Consultant project (UC) used the domain of the UNIX operating system to develop a help system that combined research in planning, reasoning, natural language processing, and knowledge representation. The user's question was analyzed and a meaning representation corresponding to the question was encoded in a knowledge representation formalism. Then, UC hypothesized the actual information needs of the user by consulting the user model and applying goal analysis. The answer was tailored to the user's expertise and goals. The sample dialogues provided were certainly impressive. However, no transcripts of real-world dialogues were provided and therefore it cannot be determined whether the methods and theories developed in UC were robust enough for practical use.

Most of the early work attempted to implement QA systems from the early perspective of AI or computational linguistics. As noted earlier, due to the limitations of the time, question answering focused on restricted domains. A turning point was reached in 1999, with the introduction of the QA track in the TREC (Voorhees 1999). The popularity of the QA track in TREC has enabled research on QA from an IR perspective. From the IR community's point of view, the task of question answering is reduced to the task of finding the text that contains the answer to the question and extracting the answer. Text documents are viewed as a source of unstructured information that is structured by indexing it. Indexing the documents makes it feasible to locate the fragments that are closely related to the question terms by applying term-matching techniques.

A consequence of this new perspective is the application of domain-independent methods, allowing what has been called **open-domain question-answering**. This approach is largely used in current QA systems. It is beyond the scope of this article to survey the techniques and systems used in open-domain QA; the interested reader is referred to the proceedings of TREC, which are available on-line.[4] Instead, in the subsequent sections we will review current approaches to question answering in restricted domains. But before that, let us analyze what a restricted domain is.

3. Characteristics of Restricted Domains

The nature of a particular restricted domain affects the kinds of questions asked and answers that can be expected. Consequently, different restricted domains benefit from different QA techniques. Some domains are particularly appropriate for the development of question answering systems. Minock (2005) lists three desiderata for a restricted

4 http://trec.nist.gov.

domain within the context of World Wide Web–based question answering—that is, question answering that relies on documents taken from the World Wide Web as the main source for finding answers. According to Minock, a restricted domain must meet the following desiderata:

1. It should be circumscribed.
2. It should be complex.
3. It should be practical.

The same desiderata apply, with some modifications, to restricted domains on question answering that is not World Wide Web–based.

3.1 Circumscription

Minock's original description of a circumscribed domain is motivated by the user's need to know what to expect of the World Wide Web–based QA system at hand and to know what questions are appropriate to the domain at hand. An example of a domain that would fare low in this desideratum is that of current events, because the user might have difficulty in ascertaining what questions can be asked. An example of a good domain according to this desideratum would be a science domain such as astronomy or chemistry.

If the QA system is not World Wide Web–based and, furthermore, is intended for use within a corporation, however, users do not face the problem of wondering what questions are appropriate. Rather, a more important motivation for a circumscribed domain is the need for clearly defined knowledge sources. The range of techniques used in a restricted domain should not need to use extensive knowledge from outside the chosen domain. Rather, a domain that has authoritative and comprehensive resources is to be preferred. Examples of resources include actual databases containing the required information.

It is natural to assume that the more restricted the domain is and the more circumscribed it is, the more possible it is to obtain such comprehensive databases. For more complex domains, useful resources are terminology databases and domain ontologies. An added value is the existence of well-accepted terminology and ontology standards.

3.2 Complexity

A domain should be complex enough to warrant the use of a QA system. This may seem an obvious statement, but it is important to bear in mind that, in a desire to find a domain that is fully circumscribed, one might attempt to develop a QA system in a domain where a simple list of facts or a FAQ would be sufficient to satisfy the user's need for information. There is no need for a QA system in such domains.

Developing a system for a simple domain does not advance research in any significant area. Such domains are to be left to those who are more interested in capitalizing on current research advances to develop practical applications, rather than extending current research boundaries. In general, the more complex a domain is, the more interesting it becomes for the researcher and the more useful it presumably is to the user.

There is a balance to be achieved between the need for a complex domain and that of a circumscribed domain, because these two desiderata are in conflict. At some point, if a domain is complex enough, it becomes difficult to manage and there is a higher probability of requiring resources belonging to other domains; in other words, the domain becomes less circumscribed.

3.3 Practicality

Practicality is an important desideratum to consider when developing a QA system. The domain should be of use to a relatively large group of people. Otherwise one risks wasting effort on a system that nobody would use, such as for an artificially constructed toy domain. The choice of domain affects the kind of users to target. Therefore, for each domain it is important to determine the kinds of questions asked in the specific domain (question style and terminology used are two important factors to consider), the sort of information that is most commonly requested, and the level of detail expected in the answers.

4. Open-Domain versus Restricted-Domain Question Answering

There are various factors that determine the best techniques to use in restricted-domain question answering, and whether techniques used in open-domain question answering would be effective in restricted-domain question answering. In this section we will briefly introduce some of these factors.

4.1 The Size of the Data

A well-known method used in open-domain QA is derived from redundancy-based techniques. These techniques were first discussed by Brill et al. (2001), who observed that, as the size of the text corpus increases, it becomes more likely that the answer to a specific question can be found with data-intensive methods that do not require a complex language model. Thus, if the question is *Who killed Abraham Lincoln?*, it is easier to find the answer in *John Wilkes Booth killed Abraham Lincoln* than in *John Wilkes Booth is perhaps America's most infamous assassin. He is best known for having fired the bullet that ended Abraham Lincoln's life*. Redundancy-based techniques are likely to have a weaker impact in restricted-domain QA, especially in the case of domains with relatively small corpora.

Domains with relatively small corpora will naturally have relatively fewer sentences that contain the answer. In those domains it becomes important to use sophisticated language processing techniques, including the resolution of inferences, if necessary, to find the answer. The haystack of a restricted domain is relatively small, but it also has fewer needles.

Note that, if the size of the corpus is relatively small, it becomes possible to apply complex NLP techniques to the complete corpus off-line. Nowadays it is possible to parse the entire corpus used in the QA track of TREC and to extract all its named entities. It is therefore feasible to parse and extract the named entities of corpora of restricted domains.

4.2 Domain Context

The actual domain provides a specific context to the question-answering process. Consequently the set of senses available to words is typically a subset of all the available senses. The impact of word-sense disambiguation is possibly reduced in RDQA, though it should be noted that some words would still have several senses available and therefore word-sense disambiguation still plays a role. We are not aware of any studies on the impact of word-sense disambiguation on restricted domains and certainly this area is worth exploring.

The kinds of questions asked in a restricted domain are naturally different from those asked in an open domain. Users of a restricted domain, and especially users who are experts in the domain, will use specific terminology and will pose rather technical questions that require very specific answers. Questions asked by such users are much more complex than those of casual users of open-domain QA systems. This is certainly the case in the medical domain, as the articles included in this special section show convincingly. The challenges related to solving those questions are certainly worth the effort in pursuing research in RDQA.

4.3 Resources

An important difference between open-domain and restricted-domain QA is the existence of specific resources for restricted domains that can be used. In the following sections we will comment on these resources.

5. Use of Domain-Specific Resources

Intuitively, a good method for answering questions in a restricted domain needs to leverage any information available about the domain in order to be able to address users' information needs with the specificity and depth required.

The type of information available for a particular domain is intrinsically related to the complexity of the domain and the particular needs of the domain users. Hence, domain knowledge representation can range from simple lists of specialized entities and terms to high-level ontologies where all the domain knowledge is unambiguously represented.

Within computer science, an **ontology** is usually defined as a formal explicit description of concepts in the domain of discourse, together with their attributes, roles, restrictions, and other defining features (Noy and McGuinness 2001). The relations between the concepts are also expressed formally. The two most common relations shown in an ontology are subclass ("is a subtype of") and instance ("is an instance of"), but other relations can be included, such as meronymy ("part of") and, in the case of WordNet (Fellbaum 1998), entailment.

For the purposes of this article, we will refer to all the possible domain knowledge representations as **ontological resources**.

Generally, domains that are complex, circumscribed, and practical are likely to have available ontological resources that can be used to quick-start QA research and development. These resources are typically developed for the domain users to help them categorize the domain knowledge and agree on notational standards, and to help them retrieve information using conventional information retrieval applications.

5.1 Open-Domain Ontologies

There are ontologies that are designed without a specific domain in mind. These are referred here as open-domain ontologies. A widely used open-domain ontology is WordNet (Fellbaum 1998). WordNet contains a large list of open-class words grouped into synonym sets (the "synsets"). A range of synset relations is encoded, such as hypernymy/hyponymy, meronymy, and entailment. WordNet also includes word relations, such as antonymy. A departure from ontologies like Cyc (Lenat and Guha 1990) is that WordNet does not include formal definitions of the features of the objects. Still, for the purposes of this article, WordNet is an ontology. This view is supported by its use in many systems, including open-domain question answering systems (Moldovan and Novischi 2002).

Open-domain ontologies like WordNet, however, are of limited use for QA in restricted domains. This is so because the information is unlikely to be well balanced with respect to the chosen domain. For example, parts of open-domain ontologies are too coarse-grained for specific restricted domains, whereas other parts are too fine-grained. And worse, open-domain ontologies may contain information that is not appropriate for specific restricted domains.

Open-domain ontologies are too coarse-grained. Restricted domains, and especially technical domains, abound in terms that are specific to the domain and largely unknown in other domains. Open-domain ontologies typically do not include these specific terms. In some domains, however, these terms may be used widely. Consequently, open-domain ontologies will need to be complemented with terminology lists or local ontologies.

Open-domain ontologies are too fine-grained. Open-domain ontologies that map words to concepts, as is the case with WordNet, face the problem of polysemous words, that is, words with multiple meanings. However, those ambiguous words are usually unambiguous in restricted domains. Take the noun *file*. WordNet 1.7.1 lists four meanings, shown in Table 1. Of the four meanings, only the first one ("a set of related records kept together") is relevant within domains related to software development. Open-domain ontologies therefore risk overloading the system with concepts that are rarely, if ever, used within the chosen restricted domain.

Open-domain ontologies may have information that is not appropriate for the domain. The most damaging property of open-domain ontologies is that they may contain information that is misleading in certain restricted domains. Restricted domains notoriously overload some terms commonly used outside their domain. For example, the usual meaning of the verb *print* is to render something into printed matter. However, within the domain of computer programming, the verb *print* usually means to display on the computer monitor. Consequently, a system that uses an open-domain ontology would possibly misinterpret the meaning of *print* in the question *Which C++ instruc-*

Table 1
Four senses of the noun *file* in WordNet 1.7.1.

Sense	*Gloss*
1.	A set of related records (either written or electronic) kept together
2.	A line of persons or things ranged one behind the other
3.	A container for keeping papers in order
4.	A steel hand tool with small sharp teeth on some or all of its surfaces; used for smoothing wood or metal

tion prints words onto the screen? This sense of *print* is not available in WordNet[5] and therefore it is not possible to apply word-sense disambiguation techniques to find the appropriate sense.

5.2 Uses of Ontological Resources

Ontological resources define a common vocabulary for accessing information in a domain and this makes it easier to manage domain information as regards the following (Noy and McGuinness 2001):

- sharing common understanding of the structure of information among people or software agents
- enabling reuse of domain knowledge
- making domain assumptions explicit
- separating domain knowledge from the operational knowledge
- making possible different analysis of the domain knowledge

Among theses concerns, enabling the separation of domain knowledge and operational knowledge is probably the most valuable characteristic for QA purposes. This fact allows the separation of the process of representing the concepts expressed in a document from the use of the relations between concepts for deduction or reasoning processes. On the other hand, formalisms, theories, and algorithms either designed for domain document representation or reasoning may be made independent from the chosen domain ontology and can also be applied to different domains, thus enhancing system portability between domains.

Research on using ontologies for QA has benefited from the following:

- The increasing availability of ontologies encoding different kinds of knowledge. We can find ontologies ranging from general world knowledge resources, such as WordNet (Fellbaum 1998), EuroWordNet (Vossen 1998), Cyc (Lenat and Guha 1990), and FrameNet (Johnson and Fillmore 2000, to very specific domain knowledge, such as the medical domain (Lindberg, Humphreys, and McCray 1993) or the chemistry domain (Barker et al. 2004).
- Steady achievements in knowledge representation and reasoning (KR&R) techniques, which enable precise representation of both domain-related information and domain-related reasoning and deduction mechanisms (Barker et al. 2004).
- Advances in the development of modular and robust natural language processing systems (Abney 1996; Hobbs et al. 1997; Basili and Zanzotto 2002) in the context of the use of ontological resources for both textual interpretation and representation (Ait-Mokhtar and Chanod 1997) and database access (Popescu, Etzioni, and Kautz 2003).

5 This was the case for the on-line version of WordNet 2.1 (http://wordnet.princeton.edu/) on 8 October 2006.

- Increasing success in the development of ontology-based QA frameworks where answers are derived from reasoning processes over questions and document ontological representations (Zajac 2001).

Ontology-based question answering systems attack the answer-retrieval problem by means of an internal unambiguous knowledge representation. Both questions and knowledge are represented using specific knowledge models based on ontological entities, concepts, and relations. The answering of questions is performed by applying different reasoning and proof techniques that allow the detection of textual entailment, which is useful in determining whether a given sentence answers a particular question.

6. The State of the Art in RDQA

Current work on QA in restricted domains tends to exploit the characteristics of the domain in order to improve the accuracy and practicability of the system. This is done largely by determining the types of information needs in the chosen domain, by studying the format of questions asked, and by leveraging the ontological information available in the domain.

Some domains are complex domains that have a history of users attempting to streamline the process to find specific information. An example of such a domain is that of medicine. It is important for a doctor to quickly diagnose the illness of a patient, and to determine if a patient is developing a new variation of an illness that has occurred before. Given the importance of finding the correct diagnosis and treatment, the domain of medicine has developed trusted resources that can be used for question answering in this domain. Zweigenbaum (2003) provides examples of resources for terminology and corpora of authoritative material.

Demner-Fushman and Lin (2005) operationalize knowledge extraction for populating a database with PICO (Population, Intervention, Comparison, and Outcome) elements from medical abstracts obtained from MEDLINE. PICO structures are the frames used for evidence-based medicine. Sang, Bouma, and de Rijke (2005a) describe several strategies for populating a database with medical information related to diseases, symptoms, and treatments, which is automatically extracted from medical texts. This structured information is used for answering medical-related questions. Niu and Hirst (2004) describe a method for identifying semantic classes and the relations between them in medical texts. This approach is able to build an ontology for the domain automatically. Yu, Sable, and Zhu (2005) present an algorithm to classify medical questions based on a well-known hierarchical evidence taxonomy (Ely et al. 2002). Rinaldi, Dowdall, and Schneider (2004) describe the difficulties in adapting an existing RDQA system developed for assisting questions on UNIX technical manuals (Mollá et al. 2000) to the Genomics domain.

Benamara (2004) reports in detail on one of the currently most advanced RDQA systems. WEBCOOP is a logic-based system that integrates knowledge representation and advanced reasoning procedures to generate responses to natural queries. This system has been developed for the tourism domain.

As for any knowledge intensive application, using ontologies for QA has as a limitation the restrictions imposed by the underlying knowledge representation models. Thus, in the following subsections we will focus on the efforts that are being employed from both historical trends in QA research (structured knowledge–based and free-text–

based perspectives) to provide systems with deep reasoning capabilities supported by ontological domain information. We introduce several works that aim at combining the use of various ontologies and we also describe current attempts to separate the processing of domain-dependent information from generic domain-independent information with the goal of increasing portability across domains.

6.1 Ontologies and Structured Knowledge–Based QA

As noted earlier, the first QA systems focused on the development of natural language interfaces to databases (NLIDBs). This is a natural approach to follow in circumscribed domains that are not very complex. The idea is to produce a structured information resource containing comprehensive information on the contents of the domain. This information resource is produced before any question is asked and is queried over when the user asks a question.

There is a wealth of research in the area of NLIDBs and it is not within the scope of this article to survey this important area of research. Rather, we refer the reader to Androutsopoulos, Ritchie, and Thanisch (1995). Work in NLIDBs assumes an existing database that is queried over. If the database does not exist, it is created by using methods based on information-extraction technology. The aim is to extract all the information that might be used as an answer. A clear candidate is the use of named entities, but the creation of templates has also been tried in open domains (Srihari and Li 2000) and restricted domains (Weischedel, Xu, and Licuanan 2004).

There are other systems that support this kind of knowledge-based question-answering, including some dealing with questions unanticipated at the time of system construction. These include the AP Chemistry question-answering system (Barker et al. 2004), Cyc (Lenat and Guha 1990), the Botany Knowledge Base system (Porter et al. 1988), the two systems developed for DARPA's High Performance Knowledge Base (HPKB) project (Cohen et al. 1988), and the two systems developed for DARPA's Rapid Knowledge Formation (RKF) project (Schrag et al. 2002).

6.2 Ontologies and Free-Text–Based QA

In this approach, users pose questions in natural language to knowledge bases made up of documents also written in natural language. In this case ontologies are used to define a language in which questions and documents can be represented and exploited to obtain the required answers. The translation from natural language to the internal representation is automatic; this presupposes fully unambiguous representations that are currently beyond our capabilities.

The main characteristic of these approaches is the intensive use of an ontology in the different parts of the question answering system. For instance, the ontology is used in the representation of the question and the documents, in the refinement of the initial query, in the reasoning processes carried out over the classes and subclasses from the ontology, and in the similarity algorithms employed for answer retrieval and extraction.

Zajac (2001) presents an ontology-based semantic framework for question answering where both questions and source texts are parsed into underspecified semantic expressions where names of the semantic atoms and predicates are defined in an interlingual ontology. Answer retrieval is done using subsumption and unification, and queries are expanded using ontological rules.

6.3 Integrating Heterogeneous Sources of Information

More interesting than using a single database is the combination of databases with semistructured information (such as text with some XML markup) or even unstructured information (i.e., plain text). This has been proposed for World Wide Web–based question answering (Lin 2002), given the availability of pockets of information stored in databases on the World Wide Web. The idea is to analyze the question and find the relevant database among a preselected list if this is possible. If there are no suitable databases or it is not possible to determine the appropriate database query, then standard question-answering techniques are applied using the World Wide Web as a resource. The same strategy can be applied to question answering over restricted domains by keeping a set of relevant databases and a corpus of documents to query over in case the question is not covered in the databases.

There are two main issues that need to be handled by a QA system that relies on heterogeneous sources:

Interface: The resources in each domain will have their own formats and interfaces, which must be unified by the QA system.

Selection: The QA system needs to determine the actual resource within which to look for the answer.

Given that the actual domain-specific resources range from simple word lists to structured databases, interfacing to them is by no means simple. Two approaches are envisaged (Lin 2002):

Slurp: Extract all the information from the multiple sources and create a database containing all the information. By having all the information in a unified database, the interfacing problem is easily solved and it is even possible to handle queries that the original databases were unable to handle (such as queries that rely on knowledge from various domains). This method is practical if the actual databases are available locally and their format is known. However, some databases are available as on-line resources only and any attempt to slurp all their information through methodical queries may be frowned upon by the database owners.

Wrap: Provide an application program interface (API) to the individual databases. The set of databases can be seen as a federated database system. The choice of providing an API has the obvious disadvantage that it may not be possible to devise a unified API that makes the best of what is available in the domain resources. The compromise would be a set of APIs that may or may not be able to query the resource with the full power of the original resource interface.

A step beyond portable QA systems is to build a meta-domain QA system. A meta-domain QA system specializes in several restricted domains by acting as a knowledge broker to specialized domain modules. An example of such a system is START (Katz 1997), which currently is a World Wide Web–based QA system that uses a wide range of structured data available on the Internet.

MOSES (Basili et al. 2004) is an ontology-based QA system in which users pose questions in natural language to knowledge bases of facts extracted from a federation of Web sites and organized in topic map repositories. This approach uses an ontology-based methodology to search, create, maintain, and adapt semantically structured World Wide Web content according to the vision of the Semantic Web in a domain related to university World Wide Web sites.

AQUA (Vargas-Vera, Motta, and Domingue 2003) combines knowledge encoded in a database with domain-related documents through an ontology that describes academic life. AQUA tries to answer a question using its knowledge base. If a query cannot be satisfied via the database, it tries to find an answer on domain-related World Wide Web pages.

The L&C system (Ceusters, Smith, and Van Mol 2003) is one of the most ambitious works in the medical domain; it tries to combine authoritative medical knowledge with information about patients. The information needed by physicians is of two sorts. First, there is information concerning patients, such as the changes in Mr. X's blood pressure over the past three days, or the substances to which Ms. Y is allergic. Second, there is what can be defined as medical knowledge, that is, the information found in textbooks, journal articles, clinical studies, and so on. The final objective of this work is to combine these two types of information so that the QA system, when asked, for example, whether it is safe to give the patient an additional shot of a hypotensive agent in order to reduce bleeding, would respond with: *Can you please wait for 45 seconds because the patient's blood pressure has been dropping slightly already for the last 2 minutes?*

6.4 Porting to Other Domains

Developing a system in a specific domain could be time-consuming. It is natural to think of ways to reuse technologies (or even code) in QA systems from other domains or from open-domain QA systems. A topic that is intimately related is that of portability to other domains.

Some question-answering systems are designed with the goals of re-usability and portability in mind. These are generic systems that can be localized to specific domains. For example, JAVELIN (Nyberg et al. 2005) is an open-domain QA system that can be extended to focus on restricted domains. Special care was taken to leverage ontologies specific to the chosen domain by developing a Java API. The specific ontological information extracted is the type hierarchy and sets of synonyms (AKA, or "also known as" extraction). Another example that demonstrates efforts in adapting an open-domain QA system to a specialized geographical environment can be seen in the work by Ferrés and Rodríguez (2006).

Another approach, developed by Frank et al. (2005), is based on the use of structured knowledge sources. This approach applies deep linguistic analysis to the question and transforms it into an internal representation based on conceptual and semantic characteristics. This representation is domain-independent and provides a natural interface to the underlying knowledge databases. This approach has been implemented as a prototype for two application domains: the domain of Nobel prize winners and the domain of language technology.

Another issue is that of localizing an open-domain QA system to a restricted domain. Nyberg et al. (2005) provides a case study that describes the problems in adapting an existing open-domain QA system to be able to deal with knowledge from existing domain ontologies.

7. Building a Restricted-Domain QA System: Main Considerations

It is difficult to imagine a general methodology for the development of an RDQA system. On the one hand, current systems are overly influenced by the specific characteristics and requirements of the domains, from the different types of questions to be answered to the heterogeneity of the knowledge available for the domain. On the other hand, the known methodological proposals (Minock 2005) are so general that they could be used to design *any* kind of information system.

Rather than propose a design methodology, we want to emphasize the main points to be taken into consideration when designing a QA system for a specific domain. These points are related to the analysis and modeling of the domain information and the selection of the appropriate technology required by the QA system. They can be listed as follows:

- domain query system analysis
- domain knowledge selection
- domain knowledge acquisition and representation
- system interface design
- technological requirements selection

Domain query system analysis: Knowing in detail all the different ways users ask for information is a prerequisite for being effective in a restricted-domain scenario. Questions need to be analyzed, classified, and associated with the different types of information the users request. The kinds of questions in a restricted domain may vary from general open-domain factoid and definition questions to very special kinds of questions that depend on the selected domain.

Domain knowledge selection: The amount and type of authoritative knowledge available for computational treatment is especially variable across different domains. For instance, there are plenty of resources for biomedical (Zweigenbaum 2003) or technical related domains, whereas, on the other hand, less popular domains (such as the legal domain) have minimal elaborated knowledge but have the advantage of enormous quantities of raw text. Domain information can be represented in different formats: from unstructured plain text documents to semi-structured (e.g., templates, SGML annotated text) or highly structured knowledge encoded in large databases and authoritative ontologies. Selecting the appropriate domain knowledge resources in each particular case is an important aspect in the design of an RDQA.

Domain knowledge acquisition and representation: Using the domain knowledge for QA purposes requires the definition of an internal representation model that allows the integration or combination of the different information sources available for the domain. The complexity of the representation model used will be proportional to the complexity of the information sources needed for encoding domain knowledge. The model selected for domain knowledge representation will also determine the kind of operational processes and reasoning techniques allowed in the domain.

System interface design: In order to obtain a natural mode of communication between users and the system, the interaction needs to be tailored according to the domain characteristics and the user requirements. Usually, natural language (NL) interfaces are preferred because they allow fluent communication between the users and the system. Nevertheless, as current natural language processing technologies do not allow the automatic translation of natural language text into a fully unambiguous content representation, NL interfaces may be sometimes substituted by template-like interfaces or unambiguous formal outputs (only useful for expert users) when exact knowledge understanding and representation is required.

Technological requirements selection: The abilities we expect from an RDQA system depend explicitly on the different aspects of the domain analysis that we have presented before. Decisions on the specific technology and methods to use will be taken according to the type of questions to be solved, the availability of specialized resources, and the representational model used for encoding the domain knowledge. As discussed in previous sections (see Sections 4 and 6), QA in restricted domains usually requires techniques that differ substantially from the techniques used in open-domain systems. Restricted domains enable the possibility of using comprehensive ontological knowledge, thus making it possible to perform more complex inferences than in open-domain QA and therefore leveraging the possibility of answering more complex questions. From this perspective taking accurate design decisions customized to the task requirements and the domain resources is essential.

8. Introduction to the Articles in this Special Section

Demner-Fushman and Lin's article (*Answering clinical questions with knowledge-based and statistical techniques*) extends previous work by the authors (Demner-Fushman and Lin 2005) on a QA system in the medical domain. The system is designed to satisfy information needs within the framework of evidence-based medicine (EBM) whereby a doctor needs to gather the current best evidence, namely, high-quality patient-centered clinical research. The data source used by the system is the set of MEDLINE abstracts, a large bibliographic database that is accessed on-line via PubMed. Input questions in this domain are highly specific and complex. Following practice in the domain, the input questions are formulated as PICO-based frames representing the major elements of a query in EBM: Problem/Population, Intervention, Comparison, and Outcome. A central task of the system is the automatic identification of PICO elements in the MEDLINE abstracts and their matching with the input query frame. In the process the system uses the Unified Medical Language System (UMLS), an extensive ontology specialized on this domain. This system is a clear example of the adaptation of the task of question answering to a specific and highly practical domain using specialized resources in order to satisfy information needs formulated as complex, structured questions.

Hallett, Scott, and Power's article (*Composing questions through conceptual authoring*) focuses on the stage of question formulation. Questions in a QA system over a specialized domain where the users are domain experts are typically complex in nature. This results in a problem both for the user, who needs to provide all the specific information in the question, and to the system that needs to analyze the question. The solution proposed in this article is to facilitate question formulation by means of Conceptual Authoring, whereby the user edits a formal representation of the query and receives feedback from an automatically generated natural language representation of that query. The article describes this method within the context of a QA system for a database of electronic health records. An analysis of the question model in this domain is presented, together with an evaluation of the usability of the method. This article presents a concept of complex query formulation that can potentially be ported to other specialized domains.

9. Conclusions

In this article we have presented an overview of methods used in QA in restricted domains and we have argued for developing research in this area. To conclude we would like to comment on two reasons for developing question answering in restricted domains:

Development of vertical systems: Restricted domains allow the development of systems that can provide the full range of processing levels and achieve a complete, end-to-end application. It therefore becomes possible to develop complete systems that can be used without the need for any time-consuming training on the methods required to formulate questions or to interpret the system results. Furthermore, restricted domains can provide a focus for the research and development of generic theories on complex question answering in particular and natural language processing in general. A clear example is the UC project developed in the 1980s. By reducing the research space it becomes possible to focus on solving complex problems that would not be attempted if the main drive was to produce a system that works in an open-domain fashion.

Applicability to current needs: General and broad scope systems are not effective in domains restricted to the interests of different kind of users: from employees in institutions and companies trying to find information in manuals and procedures, to professionals in specialized domains like law, medicine, biology, mechanics, programming, and so on. Notice that professionals in each of these areas require different types of information in their daily activities (e.g., there is a considerable difference between looking for general information on the Internet as opposed to looking for the empty weight of a wing of the Airbus A319 in a technical manual).

A major difference between open-domain question answering and restricted-domain question answering is the existence of domain-dependent information that can be used to improve the accuracy of the system. Much of the focus of this article has been on forms of tapping information from these resources.

Some domains are more appropriate for developing question answering systems than others. A domain must be circumscribed enough so that a comprehensive ontological resource can be built for the domain. A domain must be complex enough so that it presents challenging research problems in the area of natural language processing. Finally, a domain must be practical enough so that the end product is useful to a significant segment of the population. Domains (such as, for example, biomedicine) that meet all these properties are naturally more popular for researchers and developers. Consequently they have some of the best ontological information and large corpora of texts and questions that can be used for the development of such QA systems.

Question answering on restricted domains requires the processing of complex questions and offers the opportunity to carry out complex analysis of the text sources and the questions. Restricted domains also provide comprehensive ontologies and domain resources that can help in the task of processing complex questions and finding the answers. The challenges and opportunities are there for us to take.

References

Abney, S. 1996. *Part-of-speech Tagging and Partial Parsing. Corpus-Based Methods in Language and Speech*. Kluwer Academic Publishers, Dordrecht.

Ait-Mokhtar, Salah and Jean-Pierre Chanod. 1997. Incremental finite-state parsing. In *Fifth Conference on Applied Natural Language Processing (ANLP 97)*, pages 72–79, Washington, DC.

Androutsopoulos, I., G. D. Ritchie, and P. Thanisch. 1995. Natural language interfaces to databases—an introduction. *Natural Language Engineering*, 1(1):29–81.

Barker, Ken, Vinay K. Chaudhri, Shaw Yi Chaw, Peter Clark, James Fan, David Israel, Sunil Mishra, Bruce W. Porter, Pedro Romero, Dan Tecuci, and Peter Z. Yeh. 2004. A Question-answering system for AP chemistry: Assessing KR&R technologies. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Ninth International Conference (KR2004)*, pages 488–497, Whistler, Canada.

Basili, Roberto, Dorte H. Hansen, Patrizia Paggio, Maria Teresa Pazienza, and Fabio Massimo Zanzotto. 2004. Ontological resources and question answering. In *Workshop on Pragmatics of Question Answering, held jointly with NAACL 2004*, Boston, Massachusetts.

Basili, Roberto and Fabio Massimo Zanzotto. 2002. Parsing engineering and empirical robustness. *Natural Language Engineering*, 8(2/3): 97–120.

Benamara, Farah. 2004. Cooperative question answering in restricted domains: The WEBCOOP Experiments. In *Workshop on Question Answering in Restricted Domains. 42nd Annual Meeting of the Association for Computational Linguistics (ACL-2004),* pages 31–38, Barcelona, Spain.

Brill, Eric, Jimmy Lin, Michele Banko, Susan Dumais, and Andrew Ng. 2001. Data-intensive question answering. In *Proceedings TREC 2001*, number 500–250 in NIST Special Publications. NIST, pages 393–400, Gaithersberg, MD.

Ceusters, Werner, Barry Smith, and Maarten Van Mol. 2003. Using ontology in query answering systems: Scenarios, requirements and challenges. In *2nd CoLogNET-ElsNET Symposium. Questions and Answers: Theoretical and Applied Perspectives*, Amsterdam.

Chung, Hoojung, Young-In Song, Kyoung-Soo Han, Do-Sang Yoon, Joo-Young Lee, and Hae-Chang Rim. 2004. A Practical QA System in Restricted Domains. In *Workshop on Question Answering in Restricted Domains. 42nd Annual Meeting of the Association for Computational Linguistics (ACL-2004),* pages 39–45, Barcelona, Spain.

Cohen, P., R. Schrag, E. Jones, A. Pease, A. Lin, B. Starr, D. Easter, D. Gunning, and M. Burke. 1988. The DARPA high performance knowledge bases project. *AI Magazine*, 19(4):25–49.

Demner-Fushman, Dina and Jimmy Lin. 2005. Knowledge extraction for clinical question answering: Preliminary results. In *Workshop on Question Answering in Restricted Domains. 20th National Conference on Artificial Intelligence (AAAI-05),* pages 1–9, Pittsburgh, PA.

Diekema, Anne R., Ozgur Yilmazel, and Elizabeth D. Liddy. 2004. Evaluation of restricted domain question-answering systems. In *Workshop on Question Answering in Restricted Domains. 42nd Annual Meeting of the Association for Computational Linguistics (ACL-2004),* pages 2–7, Barcelona, Spain.

Doan-Nguyen, Hai and Leila Kosseim. 2004. The problem of precision in restricted-domain question-answering. Some proposed methods of improvement. In *Workshop on Question Answering in Restricted Domains. 42nd Annual Meeting of the Association for Computational Linguistics (ACL-2004),* pages 8–15, Barcelona, Spain.

Ely, J., J. Osheroff, M. Ebell, M. Chambliss, D. Vinson, J. Stevermer, and E. Pifer. 2002. Obstacles to answering doctors' questions about patient care with evidence: Qualitative study. *British Medical Journal*, 324:710–713.

Fellbaum, Christiane. 1998. WordNet: Introduction. In Christiane Fellbaum, editor, *WordNet: An Electronic Lexical Database*, *Language, Speech, and Communication*. MIT Press, Cambrige, MA, pages 1–19.

Ferrés, Daniel and Horacio Rodríguez. 2006. Experiments adapting an open-domain question answering system to the geographical domain using scope-based resources. In *11th Conference of the European Chapter of the*

Association of Computational Linguistics. Workshop on Multilingual Question Answering - MLQA'06, Trento, Italy.

Frank, Anette, Hans-Ulrich Krieger, Feiyu Xu, Hans Uszkoreit, Berthold Crysmann, Brigitte Jörg, and Ulrich Schäfer. 2006. Question answering from structured knowledge sources. *Journal of Applied Logic, Special Issue on Questions and Answers: Theoretical and Applied Perspectives*, 1:29.

Frank, Anette, Hans-Ulrich Krieger, Feiyu Xu, Hans Uszkoreit, Berthold Crysmann, Brigitte Jörg, and Ulrich Schäfer. 2005. Querying structured knowledge sources. In *Workshop on Question Answering in Restricted Domains. 20th National Conference on Artificial Intelligence (AAAI-05),* pages 10–19, Pittsburgh, PA.

Gabbay, Igal. 2004. *Retrieving Definitions from Scientific Text in the Salmon Fish Domain by Lexical Pattern Matching*. Ph.D. thesis, University of Limerick.

Galitsky, Boris. 2001a. A natural language question answering system for human genome domain. In *Proceedings of the 2nd IEEE International Symposium on Bioinformatics and Bioengineering,* Bethesda, MD.

Galitsky, Boris. 2001b. Semi-structured knowledge representation for the automated financial advisor. In *Proceedings of the Fourteenth International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, pages 874–879, Budapest, Hungary.

Green, B. F., A. K. Wolf, C. Chomsky, and K. Laughery. 1961. Baseball: An automatic question answerer. In *Proceedings Western Computing Conference*, volume 19, pages 219–224.

Hejazi, Mahmoud R., Maryam S. Mirian, Kourosh Neshatian, Bahador R. Ofoghi, and Ehsan Darudi. 2004. An ontology-based question answering system with auto extraction and categorization capabilities in the domain of telecommunications. *The CSI Journal on Computer Science and Engineering*, 2(1).

Hirschman, Lynette and Rob Gaizauskas. 2001. Natural language question answering: The view from here. *Natural Language Engineering*, 7(4):275–300.

Hobbs, Jerry R., Douglas Appelt, John Bear, David Israel, Megumi Kameyama, Mark Stickel, and Mabry Tyson. 1997. *FASTUS: A Cascaded Finite-state Transducer for Extracting Information from Natural-Language Text.* MIT Press, Cambridge, MA.

Johnson, Christopher and Charles J. Fillmore. 2000. The FrameNet tagset for frame-semantic and syntactic coding of predicate-argument structure. In Janyce Wiebe, editor, *Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics*, Seattle, WA.

Kacmarcik, Gary. 2005. Question answering in role-playing games. In *Workshop on Question Answering in Restricted Domains. 20th National Conference on Artificial Intelligence (AAAI-05),* pages 51–55. Pittsburgh, PA.

Kando, Noriko. 2005. Overview of the fifth NTCIR workshop. In *Proceedings of the Fifth NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering and Cross-Lingual Information Access*, Tokyo, Japan.

Katz, Boris. 1997. From sentence processing to information access on the World Wide Web. In *AAAI Spring Symposium on Natural Language Processing for the World Wide Web*, pages 77–94, Stanford, CA.

Katz, Boris, Sue Felshin, Deniz Yuret, Ali Ibrahim, Jimmy Lin, Gregory Marton, Alton Jerome McFarland, and Baris Temelkuran. 2002. Omnibase: Uniform access to heterogeneous data for question answering. In *Proceedings of the 6th International Conference on Applications of Natural Language to Information Systems*, pages 230–234, Stockholm, Sweden.

Katz, Boris, Jimmy J. Lin, and Sue Felshin. 2002. The START multimedia information system: Current technology and future directions. In *Proceedings of the International Workshop on Multimedia Information Systems*, Tempe, AZ.

Kim, Soo-Min and Eduard Hovy. 2005. Identifying opinion holders for question answering in opinion texts. In *Workshop on Question Answering in Restricted Domains. 20th National Conference on Artificial Intelligence (AAAI-05),* pages 20–26, Pittsburgh, PA.

Lenat, D. and R. V. Guha. 1990. *Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project*. Addison-Wesley.

Lin, Jimmy. 2002. The Web as a resource for question answering: Perspectives and challenges. In *Proceedings of the Third International Conference on Language Resources and Evaluation*, pages 2120–2127, Las Palmas, Spain.

Lindberg, D. A., B. L. Humphreys, and A. T. McCray. 1993. The unified medical language system. *Methods of Information in Medicine*, 32(4):281–291.

Minock, Michael. 2005. Where are the 'killer applications' of restricted domain question answering? In *Proceedings of the IJCAI Workshop on Knowledge Reasoning in Question Answering*, page 4, Edinburgh, Scotland.

Moldovan, Dan and Adrian Novischi. 2002. Lexical chains for question answering. In *Proceedings of the 19th International Conference on Computational Linguistics*, Taipei, Taiwan.

Mollá, Diego, Rolf Schwitter, Michael Hess, and Rachel Fournier. 2000. Extrans, an answer extraction system. *Traitement Automatique des Langues*, 41(2):495–522.

Mollá, Diego and José Luis Vicedo, editors. 2004. *Workshop on Question Answering in Restricted Domains. 42th Annual Meeting of the Association for Computational Linguistics (ACL-2004)*, Barcelona, Spain.

Mollá, Diego and José Luis Vicedo, editors. 2005. *Workshop on Question Answering in Restricted Domains. Twentieth National Conference on Artificial Intelligence (AAAI-05)*, Pittsburgh, Pennsylvania, USA.

Niu, Yun and Graeme Hirst. 2004. Analysis of semantic classes in medical text for question answering. In *Workshop on Question Answering in Restricted Domains. 42nd Annual Meeting of the Association for Computational Linguistics (ACL-2004)*, pages 54–61, Barcelona, Spain.

Noy, N. F. and D. L. McGuinness. 2001. Ontology development 101: A guide to creating your first ontology. Technical Report KSL-01-05, Knowledge Systems Laboratory.

Nyberg, Eric, Teruko Mitamura, Robert Frederking, Vasco Pedro, Matthew W. Bilotti, Andrew Schlaikjer, and Kerry Hannan. 2005. Extending the JAVELIN system with domain semantics. In *Question Answering in Restricted Domains: Papers from the AAAI Workshop*, pages 36–40, Pittsburgh, PA.

Popescu, Ana-Maria, Oren Etzioni, and Henry Kautz. 2003. Towards a theory of natural language interfaces to databases. In *Proceedings of the 2003 International Conference on Intelligent User Interfaces (IUI-03)*, pages 149–157, New York.

Porter, B., J. Lester, K. Murray, K. Pittman, A. Souther, L. Acker, and T. Jones. 1988. AI research in the context of a multifunctional knowledge base: The botany knowledge base project. Technical Report, AI-88-88, Department of Computer Sciences, University of Texas at Austin.

Rinaldi, Fabio, James Dowdall, and Gerold Schneider. 2004. Answering questions in the genomics domain. In *Proceedings of the ACL04 Workshop on Question Answering in Restricted Domains*, pages 46–53, Barcelona, Spain.

Rinaldi, Fabio, Michael Hess, James Dowdall, Diego Mollá, and Rolf Schwitter. 2004. Question answering in terminology-rich technical domains. In Mark T. Maybury, editor, *New Directions in Question Answering*. AAAI Press/MIT Press, Cambridge, MA, pages 71–82.

Rotaru, Mihai and Diane J. Litman. 2005. Improving question answering for reading comprehension tests by combining multiple systems. In *Workshop on Question Answering in Restricted Domains. 20th National Conference on Artificial Intelligence (AAAI-05)*, pages 46–50, Pittsburgh, PA.

Sang, Erik Tjong Kim, Gosse Bouma, and Maarten de Rijke. 2005a. Developing offline strategies for answering medical questions. In *Workshop on Question Answering in Restricted Domains. 20th National Conference on Artificial Intelligence (AAAI-05)*, pages 41–45, Pittsburgh, PA.

Schrag, R., M. Pool, V. Chaudhri, R. C. Kahlert, J. Powers, P. Cohen, J. Fitzgerald, and S. Mishra. 2002. Experimental evaluation of subject matter expert-oriented knowledge base authoring tools. In *Proceedings of the 2002 PerMIS Workshop: Measuring the Performance and Intelligence of Systems*, pages 272–279, Gaithersburg, MD.

Simmons, R. F. 1965. Answering English questions by computer: A survey. *Communications of the ACM*, 8(1):53–70.

Srihari, Rohini and Wei Li. 2000. Information extraction supported question answering. In *Proceedings of TREC 8 (1999)*, pages 185–196, Gaithersburg, MD.

Tsur, Oren, Maarten de Rijke, and Khalil Sima'an. 2004. BioGrapher: Biography questions as a restricted domain question answering task. In *Workshop on Question Answering in Restricted Domains. 42nd Annual Meeting of the Association for Computational Linguistics (ACL-2004)*, pages 23–30, Barcelona, Spain.

Vallin, Alessandro, Bernardo Magnini, Danilo Giampiccolo, Lili Aunimo, Christelle Ayache, Petya Osenova, Anselmo Pe nas, Maarten de Rijke, Bogdan Sacaleanu, Diana Santos, and Richard Sutcliffe. 2005. Overview of the CLEF 2005 multilingual question answering track. In *Proceedings of CLEF 2005*, Vienna, Austria.

Vargas-Vera, Maria and Enrico Motta. 2004. AQUA: A question answering system for heterogeneous sources. Technical Report KMI-04-20, KMI.

Vargas-Vera, Maria, Enrico Motta, and John Domingue. 2003. AQUA: An ontology-driven question answering system. In Mark T. Maybury, editor, *New Directions in Question Answering, Papers from 2003 AAAI Spring Symposium*, Stanford University, pages 53–57. Stanford, CA.

Voorhees, Ellen M. 1999. The TREC-8 question answering track report. In *Proceedings of TREC-8*, pages 77–82, Gaithersburg, MD.

Voorhees, Ellen M. 2001. The TREC question answering track. *Natural Language Engineering*, 7(4):361–378.

Vossen, Piek, editor. 1998. *Euro WordNet: A Multilingual Database with Lexical Semantic Networks*. Kluwer Academic Publishers, Dordrecht, Holland.

Weischedel, Ralph, Jinxi Xu, and Ana Licuanan. 2004. A hybrid approach to answering biographical questions. In Mark T. Maybury, editor, *New Directions in Question Answering*. AAAI Press/MIT Press, Cambridge, MA, chapter 5, pages 59–69.

Wilensky, Robert, David N. Chin, Marc Luria, James Martin, James Mayfield, and Dekai Wu. 1994. The Berkeley Unix Consultant project. *Computational Linguistics*, 14(4):35–84.

Woods, William A. 1997. Conceptual indexing: A better way to organize knowledge. Technical Report SMLI TR-97-61, Sun Microsystems, Inc.

Yu, Hong, Carl Sable, and Hai Ran Zhu. 2005. Classifying medical questions based on an Evidence Taxonomy. In *Workshop on Question Answering in Restricted Domains. 20th National Conference on Artificial Intelligence (AAAI-05),* pages 27–35, Pittsburgh, PA.

Zajac, Rémi. 2001. Towards ontological question answering. In *Proceedings of ACL2001, Workshop on Open Domain QA*, Toulouse.

Zweigenbaum, Pierre. 2003. Question answering in biomedicine. In *Proceedings of EACL2003, Workshop on NLP for Question Answering*, Budapest.

Appendix A: List of QA Systems in Restricted Domains

The following list is by no means exhaustive. Our purpose in presenting this list is to show the breadth of current research and applications in RDQA. We welcome updates and additions to the list, which will be maintained at http://www.ics.mq.edu.au/∼diego/answerfinder/rdqa/.

1. *Generic systems*
   - JAVELIN (Nyberg et al. 2005)
     - http://www.cs.cmu.edu/∼ehn/JAVELIN/
   - QUETAL (Frank et al. 2005, 2006)
     - http://www.dfki.de/pas/f2w.cgi?ltp/quetal-e
   - AQUA (Vargas-Vera and Motta 2004; Vargas-Vera, Motta, and Domingue 2003)
     - http://kmi.open.ac.uk/projects/akt/aqua/
     - http://kmi.open.ac.uk/projects/akt/publications.cfm
   - START (Katz 1997; Katz et al. 2002; Katz, Lin, and Felshin 2002)
     - http://start.csail.mit.edu/

2. *Collaborative learning for engineering education*
   - KAAS (Diekema, Yilmazel, and Liddy 2004)

3. *Services provided by a large company*
    - Concordia University system (Doan-Nguyen and Kosseim 2004)
4. *Salmon fish biology*
    - SOK-I (Gabbay 2004)
5. *Biography information*
    - BioGrapher (Tsur, de Rijke, and Sima'an 2004)
    - BBN Technologies (Weischedel, Xu, and Licuanan 2004)
6. *Tourism*
    - WEBCOOP (Benamara 2004)
7. *Weather forecasts*
    - System by Korea University and Sangmyung University (Chung et al. 2004)
8. *Technical domains*
    - ExtrAns (Rinaldi et al. 2004)
        - http://www.ifi.unizh.ch/cl/extrans/
    - TeLQAS (Hejazi et al. 2004)
        - http://www.neshatian.org/projects/telqas/
9. *Genomics*
    - ExtrAns (Rinaldi, Dowdall, and Schneider 2004)
    - System by KnowledgeTrail (Galitsky 2001a)
10. *Financial*
    - System by KnowledgeTrail (Galitsky 2001b)
11. *Medical domain*
    - EpoCare (Niu and Hirst 2004)
    - system by University of Maryland (Demner-Fushman and Lin 2005)
    - question classification by Columbia University and Cooper Union (Yu, Sable, and Zhu 2005)
    - IMIX
12. *Geographic domain*
    - System by UPC (Ferrás and Rodríguez 2006)
13. *Nobel prizes*
    - System by DFKI (Frank et al. 2005)
14. *Language technology*
    - System by DFKI (Frank et al. 2005)
15. *Opinion texts*
    - System by University of Southern California (Kim and Hovy 2005)
16. *Reading comprehension texts*
    - RC QA (Rotaru and Litman 2005)
17. *Role-playing games*
    - System by Microsoft Research (Kacmarcik 2005)