KNOWLEDGE MEDIA
KMi
INSTITUTE

AQUA:

A Question Answering System for Heterogeneous Sources

Tech Report kmi-04-20

Maria Vargas-Vera and Enrico Motta

The Open University

AQUA: A Question Answering System for Heterogeneous Sources

Maria Vargas-Vera and Enrico Motta
Knowledge Media Institute
The Open University
Walton Hall, Milton Keynes, MK7 6AA, UK
+44 1908 655014
m.vargas-vera@open.ac.uk

ABSTRACT

This paper describes AQUA our question answering over the Web. AQUA was designed to work over heterogeneous sources. This means that AQUA is equipped to work as closed domain and in addition to open-domain question answering. As a first instance, AQUA tries to answer a question using a Knowledge base. If a query cannot be satisfied over a knowledge base/database. Then, AQUA tries to find an answer on web pages (i.e. it uses as corpus the internet as resource). Our system uses NLP (Natural Language Processing), First order logic and Information Extraction technologies. AQUA has been tested using an ontology which describes academic life.

Keywords **Ontologies, Information Extraction, Machine Learning**

INTRODUCTION

In recent years, the rise in popularity of the web has created a demand for services which help users to skip over all irrelevant information quickly. One of the services is question answering (QA), the technique of providing answers to specific questions. Given a question such as 'which country had the highest inflation rate in 2002?' a keyword-based search engines such as Google might present the user with web pages from the Financial Times, whereas a QA system would attempt to directly answer the question with the name of a country. On the web, a typical example of a QA system is Jeeves[1] (Askjeeves 2000) which allows users to ask questions in natural language. It looks up the user's question in its own database and returns the list of matching questions which it knows how to answer. The user then selects the most appropriate entry in the list. However, users would usually prefer an answer to a given question. Therefore, a reasonable aim for an automatic system is to provide textual answers instead of a set of documents. In this paper we present AQUA a question answering system which amalgamates Natural Language Processing (NLP), Logic, Ontologies and Information Retrieval techniques to provide answers to queries in a specific domain in real time.

The first instantiation of our ontology-driven Question Answering System, AQUA (as QA-closed domain), is designed to answer questions about academic people and organizations. However, an important future target application of AQUA would be to answer questions posed within company intra-nets; for example, giving AQUA an ontology of computer systems might allow it to be used for trouble-shooting or configuration of computer systems.

AQUA is also designed to play an important role in the Semantic Web[2]. One of the goals of the Semantic Web is the ability to annotate web resources with semantic content. These annotations can then be used by a reasoning system to provide intelligent services to users. AQUA would be able to perform incremental markup of home pages with semantic content. These annotations can be written in RDF (Lassila et al. 1999, Hayes 2002) or RDFS (Brickley et al. 2000), notations which provide a basic framework for expressing meta-data on the web. We envision that AQUA can perform the markup concurrently with looking for answers, that is, AQUA can annotated pages as it finds them. In this way then, semantically annotated web pages can be cached to reduce search and processing costs.

The main contribution of AQUA is the intensive use of an ontology in several parts of the question answering system. The ontology is used

- in the refinement of the initial query
- in the reasoning process (a generalization / specialization process using classes and subclasses from the ontology), and
- in the novel similarity algorithm.

The last of these, the similarity algorithm, is a key feature of AQUA. It is used to find similarities between relations/concepts in the translated query and relations/concepts in the ontological structures. The

[1] http://www.ask.com/

[2] The goal of the Semantic Web is to help users or software agents to organize, locate and process content on the WWW.

similarities detected then allow the interchange of concepts or relations in the formulae. The ontology is used to provide an intelligent reformulation of the question, with the intent to reduce the chances of failure to answer the question. Further explanation about the similarity algorithm can be found in (Vargas-Vera et al 2003b; Vargas-Vera and Motta 2004a; Vargas-Vera and Motta 2004b)

The main contributions of this paper can be summarized as 1) To outline a proposal for Question Answering which works over heterogeneous sources (i.e. AQUA works as closed-domain and open-domain Question Answering based on information distributed on Web pages).

The paper is organized as follows: Section 2 describes the AQUA architecture. Section 3 describes AQUA as an open domain question answering. Section 4 present an scenario upon a failure to prove a question using a knowledge base. It proposes two solutions to AQUA, as open-domain question answering. Section 5 presents an evaluation of the system using confidence values associated to the rules. Section 6 describes related work closest in spirit to AQUA. Finally, section 7 gives conclusions and directions for future work.

2. AQUA ARCHITECTURE

The AQUA process model generalizes other approaches by providing a uniform framework which integrates NLP, Logic, Ontologies and information retrieval. In the AQUA architecture we identify four major phases: **user interaction**, q**uestion processing, document processing** and **answer extraction.**

1. **User interaction.** In this phase the user inputs the question and validates the answer. In a *query interface* the user inputs a question (in English) using a simple dialogue box. The user can reformulate the query if the answer is not satisfactory. The answers consist of a ranked set of answers is presented to the user for answer validation. Then, the user gives feedback to AQUA by indicating agreement or disagreement with the answer.
2. **Question processing**. Question processing is performed in order to understand the question asked by the user. This 'understanding' of the question requires several steps such as parsing the question, representation of the question and classification. The question processing phase uses the following components. Firstly, a *parser* divided a sentence its grammatical components (verb, subject, object, etc). Then the logic representation of the question is created. Secondly*, an interpreter* finds a logical proof of the question over the knowledge base using unification and resolution algorithms (Clocksin and Mellish 1981). At this stage WordNet[3] is used to find similar works and perform reformulation of the question.
3. **Document Processing**. A set of documents are selected and a set of paragraphs are extracted. This relies on the identification of the focus[4] of the question. Document processing consists of two components the first component *search query formulation* which transforms the original question, **Q** into a new question **Q'** by using transformation rules (i.e. synonymous words can be used, punctuation symbols are removed, and words are stemmed). The second component *search engine* which searches the web for a set of documents using a set of keywords.
4. **Answer processing**. In this phase answers are extracted from passages and given a score, the first component *passage selection*. This extracts passages from the set of documents likely to have the answer. Finally, the second component *answer selection* clusters answers, scores answers (using a voting model), and lastly obtains a final ballot.

We want to remind the reader that phases 1, and 2 deals with the AQUA part as closed-domain question answering. Whilst 1, 3 and 4 deals with AQUA as open-domain question answering.

In this paper we only present AQUA as a closed-domain Question Answering. Further details of AQUA as closed-domain Question Answering can be found in (Vargas-Vera et al 2003b; Vargas-Vera and Motta 2004a)

3. AQUA USED AS OPEN DOMAIN QA

The main goal of AQUA is to find an answer using heterogeneous resources. However, this paper focuses on the problem of AQUA as an open-domain. It generates a proof over a knowledge base but if the proof fail. Then, AQUA tries to find an answer using documents on the internet. Our domain of study was as first instance, academic life. Therefore, in our scenario we want to prove for example that X works on Y where X is a researcher and Y is a project. This query is translated into FOL (First Order Logic) predicate works(X,Y) which can be true or false. If false then the prove uses documents which satisfy rules about each concept in the query in this particular example **researcher** and **project**.

[3] http://www.cogsci.princeton.edu/~wn/

[4] Focus is a word or a sequence of words which defines the question and disambiguates it in the sense that it indicates what the question is looking for.

In our scenario, academic life, we might be interested to prove that an specific researcher works on an specific project. Therefore, there are two entities which should be extracted from web pages. These pages are located having a rule based system which define concept researcher and concept project. This information extraction process relies on the structure of documents provided by HTML format. Our system has been implemented in Sicstus Prolog and was translated into JIP Prolog (Chirico 2003) and POW (Kushmerick 2003).

4. SCENARIO

4.1 SOLUTION 1 – RESTRICTED INFORMATION EXTRACTION (IE)

We have implemented a rule based system. Each of the rules describe the properties of each concept which will be extracted. In our domain (academic life) we are interested in extracting projects, publications, research areas, researchers and technologies. An example of a rule for instance researcher is as follows:

researcher(Researcher):- websearchmatchc(['site:portal.acm.org',Researcher],[Researcher,'</title>']).

researcher(Researcher) :- websearchmatchc(['site:portal.acm.org.portal.cfm',Researcher],['research page','</title>']).

A working verb for instance is the verb **work.** The rule in Prolog notation is given below.

workingverb(Verb) :- Verb = work ; Verb = associated.

The concepts which will be extracted from documents are defined with the predicate my_roles.

my_roles :- asserta(roles([researcher,project,workingverb])).

The training is performed providing some examples such as the given below.

my_example :- asserta(example('Fensel works Akt.', ['Fensel','Akt', works])).

my_example :- asserta(example('Maria Vargas-Vera works in Akt project.', ['Maria Vargas-Vera','Akt', works])).

The deduction process is performed using the rules obtained during the training phase. Currently, the rules for extraction are simple regular expressions. However, in future, we would like to have more sophisticated regular expressions.

Let us imagine that we pose the question **Does Fensel work on Akt?**

The query is translated in FOL predicate

work(fensel,akt)

By using the KMi ontology our query is reformulated into

work(fensel,akt) & project(akt)

AQUA attempts to satisfy the predicates on the knowledge base. Let us imagine the scenario where the proof had failed. Then, a request is sent to a Prolog program to collect WWW documents which can be used to satisfy our initial query. A summary of the process can be found as follows:

There are two cases:

1. The system uses as facts (in Prolog sense) the examples used during training phase. If Fensel was used as example of researcher then the system will deduce straightforward that **Fensel is a researcher**

   * verifying "Fensel works Akt"

   \+ verifying that the pnseq[5] "Fensel" is a researcher

   * Proved that Fensel is a researcher by looking up an example

   As second step the system shout verify that **akt is not a researcher** and that **akt is a project**.

   * Trying to prove that Akt is a project by trawling with [Fensel, Akt, works]

2. There is not example stating that **Fensel is a researcher** then

   * verifying "Fensel works Akt"

   \+ verifying that the pnseq "Fensel" is a researcher

   * Trying to prove that Fensel is a researcher by trawling with [Fensel, Akt, works]

   ** Mirror.insert: cache hit for URL http://kmi.open.ac.uk/publications/papers/kmi-tr-151.pdf [id m486]

   ** Mirror.insert: cache hit for URL http://kmi.open.ac.uk/publications/papers/Kmi-tr-106.pdf [id m489]

   ** Mirror.insert: cache hit for URL http://www.hltcentral.org/page-984.shtml [id m490]

   ** Mirror.insert: cache hit for URL http://eprints.aktors.org/archive/00000125/01/brewster_nldb02.pdf [id m491]

---

[5] pnseq(S,T) unifies T with each maximal sequence of proper nouns in sentence S. S must be bound to an atom or a string.

** Mirror.insert: cache hit for URL http://www.idi.ntnu.no/~brase/pub/nldb-brase-gullaV40.pdf [id m122]

…….

The system tries to establish that **Fensel is a researcher**. Then, POW tries to prove that Akt is a researcher. This seems to be one of the main limitations of POW. Therefore, we believe that POW should deal with constraints over variables used in learnt patterns. The lack of constraints leads to extra overload. For example, the search that **Akt is a researcher** could be avoided.

+ verifying that the pnseq "Akt" is a researcher.

We argue that by restricting the type of arguments. The search space could be reduced. Then extra overhead could be avoided.

Coming back to our example, the proof continues until it is proved that **Fensel does not work on Akt**.

In conclusion we can say that solution 1 does not require annotating documents for training. However, it searches (n*n) times where n is the number of concepts to be extracted from documents.

4.2 SOLUTION 2:NON RESTRICTED INFORMATION EXTRACTION

A second solution implemented using 3 components from UMass (Marmot, Badger and Crystal). This solution requires annotation of documents for the training phase. We used an electronic newsletter as corpus. This newsletter describe events happening in an academic institution KMi (short name for Knowledge Media Institute). The kind of questions we posed to AQUA are the ones presented in Table I. In a first instance, we asked questions about **visitors, academic-conferences and awards**. The patterns are shown in Table I.

KMi News articles corpus

Visits patterns	**Questions**
X visited Y	Who visited Edinburgh University?
Y was visited by X	Does OU was visited by Mr Blair ?
Y visited by X on Z	Who visited the OU on 12 July 2003?
Y were visited by X	Who were visited by Mr Davis?
X came to visit Y	Who came to visit The Open University?
Y hosted X	Who hosted Ernesto Compatangelo?
Y hosted a visit from X	Who hosted a visit from Ernesto Compatangelo?
Y had a visit from X	Who had a visit ?
Y welcomes X	Who welcomes Gilliam Vincent?
In all patterns shown above X is a person, Y is a place/institution and Z is a location.	
Awards patterns	
ORG has been awarded MONEY by FUNDER	Who was awarded 10000 Euros by the European Commission ?
ORG has been received MONEY from FUNDER to carry out research in TOPIC	Which organization has been awarded 10000 Euros from the EC to carry out research in the semantic web?
ORG has been awarded MONEY by FUNDER to carry out research in TOPIC	Which organization has been awarded 10000 Euros by EC to carry out research in the semantic web?
ORG has been awarded MONEY on TOPIC	Which organization has been awarded 10000 Euros by EC to carry out research on the semantic web?
awards MONEY to ORG	
FUNDER has awarded MONEY to ORG	What funder has awarded 10000 Eu to Kmi?
brings MONEY to ORG	Who brings money to ORG?
Academic conference patterns	
held at PLACE	What was held at the Open University?
taking place in PLACE	Where did the meeting take place?
hosted by ORG	Who hosted the conference?
organized by ORG	Who organized the conference?
paper entitled TITLE	What was the title of the

	paper?
Work/Associated to a project	
X work Y	Does X work on Project Y?
X associated Y	Does X is associated to Y

Table I shows a list of patterns and questions

In this second solution we found that the main problem was that spurious rules were generated by Crystal (a learning component). A first round of experiments showed that an automatic mechanism is needed in order to determine which extraction rules are spurious. As first, we believed that this problem could be solved by associating confidence value to the extraction rules (Vargas-Vera and Celjuska 2003;). Previous work has reported that spurious patterns were deleted manually from the library of rules under the assumption that they were not likely to be of much value (Riloff 1996a). We confirmed our expectations of getting rid of spurious rules by associating confidence to the rules automatically (Celjuska and Vargas-Vera 2004).

One of the main drawbacks of solution 2 is that the corpus need to be annotated by a user. However, extraction is faster than in solution 1.

5. EVALUATION

We focused on associating a confidence value with the Crystal-induced rules (Marmot 2001) in order to get rid of spurious rules automatically. The confidence value for each rule was computed by a five-fold cross-validation methodology on the training set. According to this methodology, the training set is split into five equally sized subsets and the learning algorithm is run five times. Each time, four of the four pieces are used for training and the fifth is kept as unseen data (test set) for the evaluation of the induced rules. The final result is the average over the five runs. At run time, each instance extracted by Badger (Marmot 2001) will be assigned the precision value of that rule. The main feature of using confidence values is that, when presented with ambiguous instantiations, we can still choose the one with the highest estimated confidence.

5.1 VALIDATION OF THE SYSTEM AND EXPERIMENTS

The corpus used was an electronic newsletter containing news articles describing events. This news articles were short text articles, were gathered from five different archives. Overall we used 300 news articles in our experiments. Manual annotation was needed for each of the representative news articles of each event.

The experiments conducted showed that using the rule confidence might increase the precision by around 15% depending on different models and parameters (Celjuska and Vargas-Vera 2004). In addition, it was observed that using the 5-Fold-Cross methodology for its computation seems to be a better choice to the simple method of taking **Coverage and Error** values computed by the learning component Crystal (Marmot 2001).

6. RELATED WORK

There are many trends in question answering (Katz 1997, Katz et al. 1988a, Katz 1988b, Plamondon et al. 2001, Burke et al. 1997, Moldovan 1999, Hovy et al. 2001a, Hovy et al. 2001b,Attardi 2001). However, in this paper, we only describe the systems most closely related to the AQUA system philosophy.

MULDER is a web-based QA system (Kwok 2001) that extracts snippets called summaries and generates a list of candidate answers. However, unlike AQUA, the system does not exploit an inference mechanism, and so, for example, cannot use semantic relations from an ontology.

QUANDA is closest to AQUA in spirit and functionality. QUANDA takes questions expressed in English and attempts to provide a short and concise answer (a noun phrase or a sentence) (Breck et al. 99). Like AQUA, QUANDA combines knowledge representation, information retrieval and natural language processing. A question is represented as a logic expression. Also knowledge representation techniques are used to represent questions and concepts. However, unlike AQUA, QUANDA does not use ontological relations.

ONTOSEEK is an information retrieval system coupled with an ontology (Guarino 1999). ONTOSEEK performs retrieval based on content instead of string based retrieval. The target was information retrieval with the aim of improving recall and precision and the focus was specific classes of information repositories: Yellow Pages and product catalogues. The ONTOSEEK system provides interactive assistance in query formulation, generalization and specialization. Queries are represented as conceptual graphs, then according to the authors "the problem is reduced to ontology-driven graph matching where individual nodes and arcs match if the ontology indicates that a subsumption relation holds between them". These graphs are not constructed automatically. The ONTOSEEK team developed a semi-automatic approach in which the user has to verify the links between different nodes in the graph via the designated user interface. In contrast, AQUA does not require user intervention in building the **graph of the query** and the **graph obtained from the ontology.**

7. CONCLUSIONS AND FUTURE WORK

In this paper we have presented AQUA - a question answering system which merges NLP, Logic, Information Retrieval techniques and Ontologies[6]. AQUA translates English questions into logical queries that are then used to generate of proofs. Currently, AQUA is coupled with the AKT reference ontology for the academic domain. In the future, we plan to couple AQUA with a set of ontologies from our repertoire of ontologies.

AQUA (as closed-domain) makes use of an inference engine which is based on the Resolution algorithm. However, in future it will be tested with the Contextual Resolution algorithm which will allow the carrying of context through several related questions.

One of the main restrictions of AQUA is that it only can answer questions in a isolation (i.e. it does not handle a set of follow up questions related to the initial question). Therefore, we will explore how to handle context across a set of queries.

Finally, future research can be focused one of the main problems found in AQUA. One such problem is that it needs to learn paraphrases to improve the QA process. For instance AQUA could learn that X killed Y has the same linguistic meaning that Y was assassinated by X (Duclaye et al. 2002).

Our current work is focusing on embedding AQUA in the AKT semantic portal which will offer a variety of services to support academics in knowledge intensive tasks. A semantic portal can be seen as an entry point to knowledge resources that may be distributed across several locations. Such portal should provide means for navigating through all these resources in an easy way, since it may be used by users with all levels of computing knowledge. This semantic portal will offer users services such as semantic browsing, smart question answering and ontological browsing (Stojanovic *et al.*, 2001; Studer *et al.*, 2002; Moreale and Vargas-Vera 2003).

ACKNOWLEDGMENTS

This work was funded by the Advanced Knowledge Technologies (AKT) Interdisciplinary Research Collaboration (IRC), which is sponsored by the UK Engineering and Physical Sciences Research Council under grant number GR/N157764/01. The authors of this paper want to thank David Celjuska for the experiments performed using Marmot, Crystal and Badger.

---

[6] AQUA has been implemented in Sicstus Prolog, C, OCML, POW and PHP.

REFERENCES

Askjeeves: http://askjeeves.com/, 2000.

Attardi G. and Cisternino A. and Formica F. and Simi M. and Tommasi A. (2001): Proceedings of TREC-9 Conference, NIST, pp 633-641, 2001.

Burke R. D. and Hammond K. J. and. Kulyukin V. A. and Lytinen S. L. and Tomuro N. and Schoenberg S. (1997): Questions answering from frequently-asked question files: Experiences with the FAQ Finder System. The University of Chicago, Computer Science Department, TR-97-05, 1995.

Brickley D. and Guha R.(2000): Resource Description Framework (RDF) Schema Specification 1.0. Candidate recommendation, World Web Consortium, URL:http://www.w3.org/TR/2000/CR-rdf-schema-20000327.

Breck E. and House D. and Light M. and Mani I. (1999): Question Answering from Large Document Collections, AAAI Fall Symposium on Question Answering Systems, 1999.

Celjuska D. and Vargas-Vera M. Ontosophie: A Semi-Automatic System for Ontology Population from Text. KMi-TR-155, The Open University, August 2004.

Chirico U. (2003) JIProlog 2.0 Reference Manual. http://www.ugosweb.com/jiprolog/

Clocksin W. F. and Mellish C. S. (1981): Programming in Prolog, Springer-Verlag. 1981

Doan A. and Madhavan J. and Domingos P. and Halevy A. (2002): Learning to Map between Ontologies on the Semantic Web. In Proc. of the 11th International World Wide Web Conference (WWW2002), 2002.

Duclaye F. and Francois Y. (2002): Using the Web as a Linguistic Resource for Learning Reformulations Automatically , Proceedings of the third international conference on language resources and evaluation (LREC'02) vol. 2, pp. 390-396, Las Palmas, Canary Islands - Spain, May 2002.

Frakes W. and Baeza-Yates R. (1992): Information Retrieval: Data Structures & Algorithms, Prentice Hall, 1992.

Guarino N. (1999): OntoSeek: Content-Based Acess to the Web, IEEE Intelligent Systems, pp 70-80, 1999.

Hayes P. (2002): RDF Model Theory, W3C Working Draft, URL:http://www.w3.org/TR/rdf-mt/

Hovy E. and Gerber L. and Hermjakob U. and Junk M. and Liu C-Y (2001a): Question Answering in Webclopedia, Proceedings of TREC-9 Conference, NIST, 2001.

Hovy E. and Gerber L. and Hermjakob U and Liu C-Y and Ravichandran D. (2001): Toward Semantics-Based Answer Pinpointing, Proceedings of DARPA Human Language Technology conference (HLT), 2001.

Katz B. (1997): From sentence processing to information access on the world wide web, Proceedings of AAAI Symposium on Natural Language Processing for the World Wide Web, 1997.

Katz B. and Levin B. (1988a): Exploiting Lexical Regularities in Designing Natural Language Systems, MIT Artificial Intelligence Laboratory, TR 1041, 1988.

Katz B. (1988b): Using English for Indexing and Retrieving, MIT Artificial Intelligence Laboratory, TR 1096, 1988.

Kushmerick N. (2003): POW Prolog on the Web. Version pow-2.0.Computer Science, Department, University College Dublin, 2003.

Kwok C. and Etzioni O. and D. S. Weld D. S. (2001): Scaling Question Answering to the Web, World Wide Web, pp 150-161, 2001.

Lassila O. and Swick R. (1999): Resource Description Framework (RDF): Model and Syntax Specification. Recommendation. World Wide Web Consortium, URL: http://www.w3.org/TR/REC-rdf-syntax/.

MARMOT User Guide (2001): Natural Language Processing Laboratory, Center for Intelligent Information Retrieval, University of Massachusetts, Amherst, USA. Task Domain Specification and User Guide for Badger and Crystal. Natural Language Processing Laboratory, Center for Intelligent Information Retrieval, University of Massachusetts, Amherst, USA.

Moldovan D. and Harabagiu S. and Pasca M. and Mihalcea R. and Goodrum R. and Girju R. and Rus V. (1999): LASSO: A Tool for Surfing the Answer Net. Proceedings of TREC-8 Conference, NIST, 1999.

Moreale E. and Vargas-Vera M. (2003): Towards a Student Portal. KMI-TR-136, Knowledge Media Institute, The Open University, 2003.

Plamondon L. and Lapalme G. and Diro R. and Kosseim L (2001): The QUANTUM Question Answering System, Proceedings of TREC-9 Conference, NIST, 2001.

Pulman S. G. (2000): Bidirectional Contextual Resolution, Computational Linguistic Vol 26/4, 497-538, 2000.

Ravichandran D and Hovy E. (2002): Learning Surface Text Patterns for a Question Answering System. In Proceedings of the ACL Conference, 2002.

Riloff E. (1996b): Automatically Generating Extracting Patterns from Untagged Text. Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96)}, pp 1044-1049, 1996.

Stojanovic, N., Maedche, A., Staab, S., Studer, R. and Sure, Y. (2001). SEAL – A Framework for Developing Semantic PortALs, *K-CAP'01*, October 22-23, 2001.

Studer, R., Sure, Y. and Volz, R. (2002). Managing User Focused Access to Distributed Knowledge, *Journal of Universal Computer Science* (*J.UCS*), 8, 6, 662-672, 2002.

Vargas-Vera M. and Celjuska D.. Event Recognition on News Stories and Semi-Automatic Population of an Ontology. IEEE/ACM International Joint Conference on Intelligent Agent and Web Intelligence (WI 2004), Beijing, China, September 20-24 2004, publisher IEEE Computer Society Press, 2004.

Vargas-Vera M. and Motta E. (2004b): An Ontology-driven Similarity Algorithm. KMI-TR-151, Knowledge Media Institute, The Open University, July 2004.

Vargas-Vera M. and Motta E. (2004a): AQUA - Ontology-based Question Answering System. Accepted to be presented at Third International Mexican Conference on Artificial Intelligence (MICAI-2004), LNAI 2972 Advances in Artificial Intelligence, Springer Verlag, Editor R. Monroy. April 26-30, 2004.

Vargas-Vera M. and Celjuska D. (2003): Ontology-Driven Event Recognition on News Stories. KMI-TR-135, Knowledge Media Institute, The Open University, 2003

Vargas-Vera M. and Motta E and Domingue J. (2003a): AQUA: An Ontology-Driven Question Answering System. AAAI Spring Symposium, New Directions in Question Answering, Stanford University, March 24-26, 2003.

Vargas-Vera M. and Motta E. and Domingue J. (2003b): An Ontology-Driven Question Answering System (AQUA). KMI-TR-129, Knowledge Media Institute, The Open University, 2003.