

Aqua fact-file

Owner	: Open University
Researchers (listed alphabetically)	: Martin Dzbor [Browse, RDF] , Dr E Motta [Browse, RDF] , Maria Vargas-Vera [Browse, RDF]
Description	: http://kmi.open.ac.uk/projects/akt/aqua/
Builds on	: PHP , SICStus Prolog , C
Addresses challenges	: Knowledge Retrieval

What's the Problem?

The use of the web has become popular and also the need of services that could exploit the vast amount of information in it. Therefore, there is a need for automated question answering systems. These kind of systems should allow users to ask questions in everyday language and receive an answer quickly and with a context which allows the user to validate the answer.

AQUA is a tool which give an answer to specific question written in English as opposed to document retrieval.

Towards a Solution

We had proposed an architecture for the AQUA system which amalgamate different perspectives: Natural Language Processing (NLP), Logic and Ontologies.

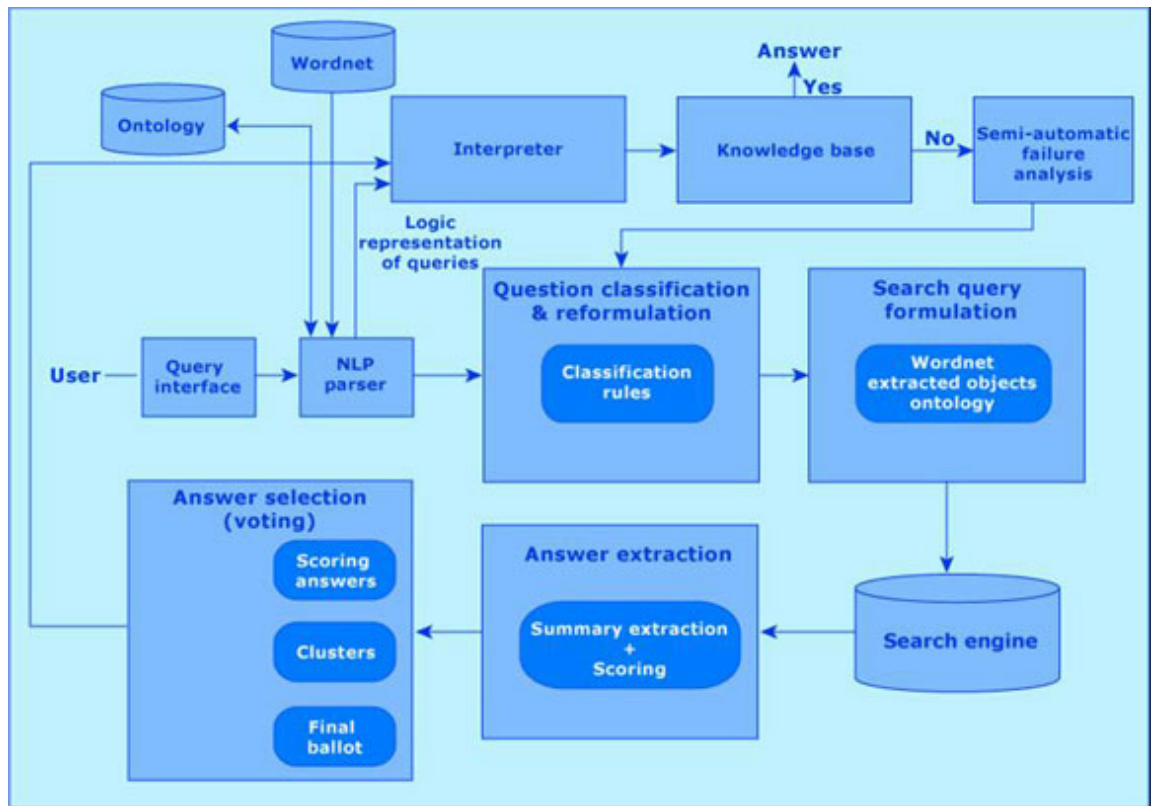
The AQUA architecture

Figure below shows the ideal architecture of our AQUA system. Each module in the architecture is described as follows:

- **Query interface.** The user writes his/her question using the query interface. This query interface is a Google sort type interface. If the user does not obtain a satisfactory answer then he/she could reformulate the query.
- **The NLP parser** does the segmentation of the sentence into subject, verbs, prepositional phrases, adjectives and objects. The output of this module is the logic representation of the query.
- **WordNet.** It is used as dictionary in the AQUA system.
- **Ontology.** We use a hand-crafted ontology containing people, projects, publications, technologies and events.
- **Knowledge base.** This knowledge base is constructed incrementally and it is domain specific. In our case is knowledge base containing information about our organization such as researchers, projects, publications, technologies and events happening in our institute.
- **Interpreter** is the logic interpreter which executes a query using unification and resolution algorithms. It finds a proof of the query against the knowledge base.
- **Failure analysis.** This subsystem analyzes the failure of a given question and gives an explanation why the query failed. Then the user could provide new information for the proof. At this point the proof could be re-assume. This process could be repeated several times as is needed.
- **Question classification & reformulation** classifies question as belonging to any of the types supported in AQUA (what, who, when, which, why and where).
- **Search query formulation.** In this module we transform the original question using transformation rules into a new question Q'. At this stage synonymous words are used,

punctuation symbols are removed and words are stemmed.

- **Search engine** searches in the web for a set of documents which satisfy the query using a selected set of keywords.
- **Answer extraction** extracts information from the set of documents that the search engine found satisfying the question Q'.
- **Answer selection** it has three functionalities. It clusters answers, scores them using the voting model and finally it obtains a final ballot.

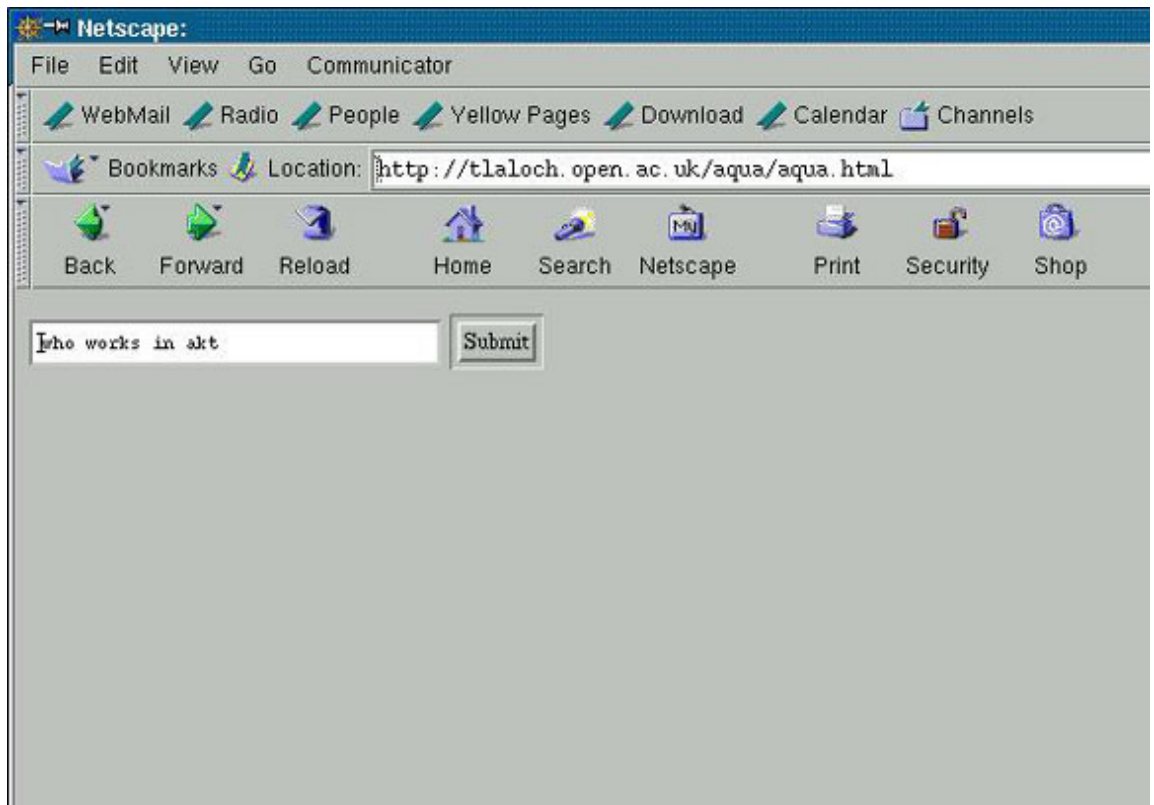


Some examples of the typical questions in the AKT portal are as follows:

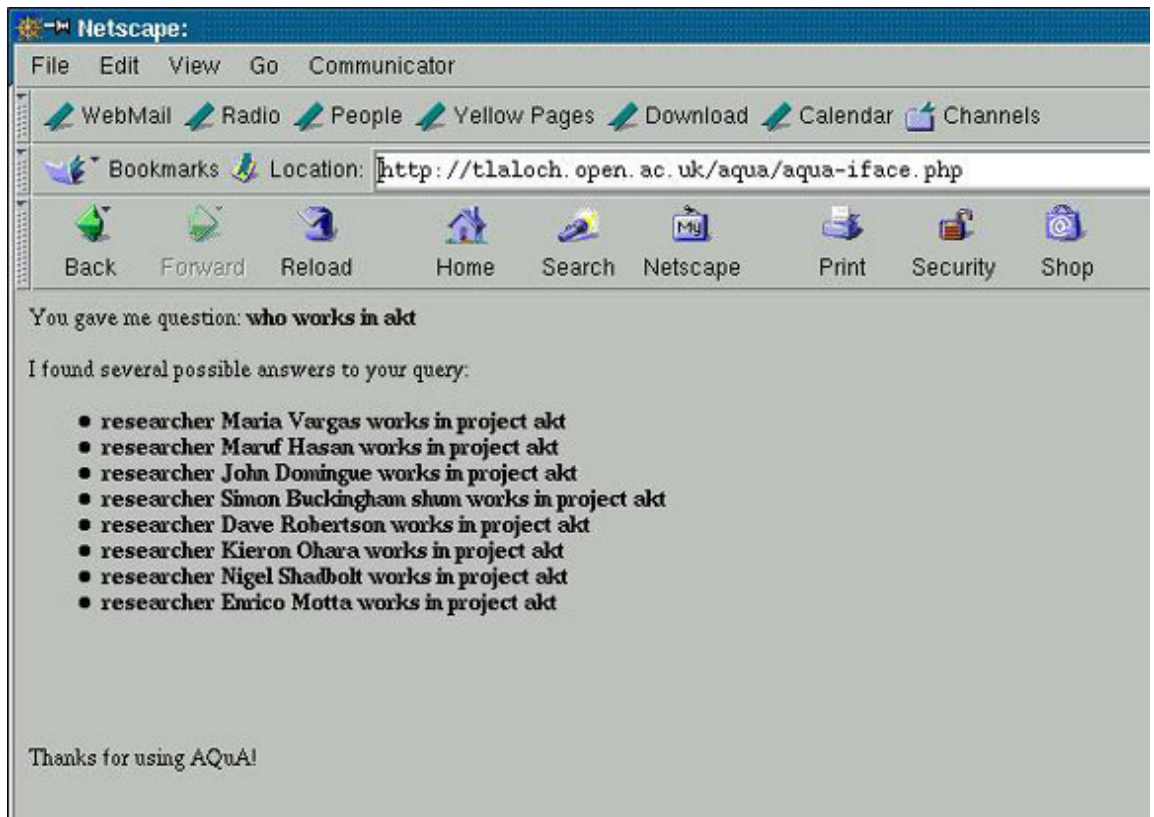
- Who works in the akt?
- Which technologies institution kmi had produced?
- Who works in ontologies?
- Who owns mnm?

Demo

The user writes the question in English:



Then AQUA produces the following set of answers:



Technical requirements: Netscape/Internet Explorer browser

Further reading:

AQUA: An Ontology-Driven Question Answering System. Maria Vargas-Vera, Enrico Motta and John Domingue. AAAI Symposium on New Directions of Question Answering, Stanford University, 24-26 March, 2003. [PDF](#)

Semantic representation

View in the [AKT Triplestore Browser](#) or as [RDF](#).

Also available in [DOAP RDF](#) ([Description Of A Project](#))