2009 Ninth International Conference on Intelligent Systems Design and Applications

# CASTALIA: Architecture of a Fuzzy Metasearch Engine for Question Answering Systems

Jesus Serrano-Guerrero
Jose A. Olivas
Jesus A. Gallego
Francisco P. Romero
University of Castilla-La Mancha
Dpt. of Information Technologies and Systems
Paseo de la Universidad 4, Ciudad Real, Spain, 13071
{jesus.serrano, joseangel.olivas, franciscop.romero} @uclm.es

Andres Soto
Department of Computer Science
Universidad Autónoma del Carmen
Ciudad del Carmen, Campeche, México, 24160
soto_andres@yahoo.com

## Abstract

*The goal of this paper is to present the architecture of a metasearch engine called Castalia, still under development, which includes several underlying Q&A systems. Usually metasearch engines manage typical search engines like Google or Yahoo, but in this case the encapsulation of Q&A systems proposes new challenges that can be modeled by fuzzy logic apart from the other existing challenges such as the fuzzy modeling of temporal or causal questions.*

## 1 Introduction

Nowadays a huge growth of information is taking place. It is estimated that the most popular search engines such as Google, Yahoo or Altavista are able to access over 30% of the information from the Web. But not only the Web can store that information, also the companies or the goverments need to store knowledge that they are not capable of retrieving later.

Therefore a large volume of information exists but it is so bad-structured that the access to certain pieces of information becomes an extremely complicated task. This problem involves the necessity of developing specialized techniques to deal with these information needs.

Hence the automatic information processing systems arose with the aim of providing an effective and efficient manner to retrieve information for the users needs. The main differences between the different types of systems rely on the way of processing the information but above all, the final objective of the information.

Several types of automatic information processing systems can be found such as Information Retrieval Systems, Information Extraction Systems or Q&A Systems.

The objective of an Information Retrieval System is to identify documents within a collection that are relevant to a user's information request. As a result, these systems return a ranked document list, whereas Information Extraction Systems try to process documents in order to find and to extract relevant knowledge to the user. And finally the aim of the Q&A Systems is to answer concrete user questions.

Q&A systems are a special class of Information Retrieval Systems, where the system should be able to answer questions posed in natural language from document collections. These search collections could include local document collections as well as the World Wide Web.

On the one hand Q&A systems could be classified in:

- Closed-domain: which are oriented just to answer questions about some specific domain.
- Open domain: which are supposed to answer questions about any topic.

978-0-7695-3872-3/09 $26.00 © 2009 IEEE
DOI 10.1109/ISDA.2009.104

And on the other hand according to the Roadmap Committee [1], the Q&A systems can be classified taking into account several research issues. Among them we highlight the following:

- Question classes: Different types of questions require the use of different strategies to find the answer.
- Question processing: The same information request can be expressed in different ways.
- Context and Q&A: Questions are usually asked within a context and answers are provided within that specific context.
- Data sources for Q&A: Before a question can be answered, it must be known what knowledge sources are available.
- Answer formulation: The result of a Q&A system should be presented in a way as natural as possible.
- Multi-lingual question answering: The ability of developing Q&A systems for other languages, not only English.

Throughout history many Q&A systems have been developed, for example, BASEBALL [9]. BASEBALL answered questions about the baseball league in the U.S.A. One of the possible problems of this type of systems is they tend to answer questions from a single domain, that is, a different system exists for each domain with a different interface to answer the questions.

However often the question answers are stored in distributed sources. This is the reason why metasearch engines arose. When a metasearch receives a user request, it selects the best search engines for that query and executes the query in all of them. The great advantage of metasearch engines is they provide a common interface for users, i.e., a user is using several information sources jointly by means of a unique interface in an easy manner.

Metasearch engines commonly are used for grouping several typical search engines such as Google or Yahoo but this paper wants to present a system that goes beyond this idea.

Castalia project tries to answer the question formulated by Olivas [14]: 'MetaQAS, why not?'. Castalia intends to encapsulate Q&A systems developing a metasearch engine based on fuzzy logic which will perform questions on multiple domains using a single interface to formulate them.

This approach presents several new challenges for the Web, apart from the existing [13, 6], that will be commented below.

The remainder of the paper is organized as follows: next section presents the different point in which fuzzy can work successfully with respect to the Q&A systems. Section 3 explains the Castalia architecture in general whereas Section 4 and 5 explains the two main modules of this architecture in detail. Finally some conclusions and future works will be pointed out.

## 2 Fuzzy logic and Q&A systems

Several new proposals are appearing based on fuzzy logic which try to manage the information stored by information retrieval systems. Several of these new proposals focus all their attention on answering natural language questions.

There are many types of question classifications according to different criteria. For example, questions can be explicit or implicit, and the explicit question can be classified into two main types: Yes-no questions and Wh-questions.

Yes-no questions are characterized by using as main verb "to be" or "to have" also they can begin with auxiliary or modal verbs (can, could, do, does, etc.) whereas Wh-question are characterized by starting with the terms: when, who, what, which, how, etc.

There are a lot of kinds of Q&A systems whose performance can vary depending on the type of the asked question. Not all of them are always able to return an exact answer and they may return a set of links or definitions[1] where interesting related information may be found.

Two kinds of answers can be considered: hard and soft answers. Hard answers are simple answers that do not provide any additional information. For example, the answer for a temporal question can be a date, a day or even a time period; or the answer for a spacial question can be a city, a country or an address. On the other hand, soft answers provide additional information that can complete the answer. For example, there are Q&A systems specialized in questions asking for a concrete individual, i.e., they can be answered by a single name ( WHONAMEDIT[2]):

Question: Who invented the radio?

Answer: Guglielmo Marconi

or they can be answered by a bibliographical answer

Question: Who is Martin Luther?

Answer:

- Born: 15 January 1929
- Birthplace: Atlanta, Georgia
- Died: 4 April 1968

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

[2] http://www.whonamedit.com/

- Best Known As: The civil rights hero who said “I have a dream”

> Martin Luther King, Jr. was an African-American clergyman who advocated social change through non-violent means. A powerful speaker and a man of great spiritual strength........

As can be seen in this example, asking for an individual, additional information can be useful for specifying the final answer or for avoiding inconsistent answers.

Usually Q&A systems provide soft answers which tend to explain in detail the answer.

New challenges can appear from this idea. For example, all texts are not susceptible to be represented by fuzzy sets therefore it is necessary to design filters whose function is to detect this type of texts.

Each answer provided by an Q&A system may not be complete and it could be completed with the information provided by other Q&A systems. Thus, two answers from two different Q&A systems for the question ‘*what time did Robert arrive?*’:

> Robert arrived too late.
>
> Robert arrived at 5.00 am.

They can be complementary and it may be possible to merge both into a single answer:

> Robert arrived late, at 5.00 am.

The concept *late* is very difficult to be interpreted, however, several attempts can be found which try to manage adverbs by means of fuzzy logic [23] achieving successful results. On the other hand the time *5.00 am* is a hard answer that is not easily interpreted as a fuzzy concept. However both answers can work jointly as fuzzy sets because it can be interpreted that there exists a time span where the time *5.00 am* is not included. Now the problem is to achieve that time span from the additional information of the retrieved answers. For time intervals it is necessary to define a starting point and an ending point but it is not always possible to establish these time spans due to lack of information.

Also the fuzzy representation of the time spans can be useful as filter to select or reject answers. For instance, if two answers are retrieved, one of them is a time span and the other one is a year not included into that time span, then both queries are inconsistent and at least one of them should be wrong.

In the same way as the adverbs have been handled by fuzzy logic, the events have been studied in order to be represented by fuzzy logic as well [18, 2].

As many other information retrieval systems, a simple question is not descriptive enough then it is necessary to complete its meaning adding new related terms or even generating new alternative questions expressing the same idea. A lot of fuzzy logic-based techniques have been proposed to deal with this problem [4, 20, 11] but it is not only necessary to create new queries but these have to be able to retrieve answers potentially interpretable by fuzzy logic. Therefore it is not only necessary to achieve new information but it is also necessary to guide the search process in order to retrieve information that can be modeled in a fuzzy way. Hence it is necessary to define measures to assess the similarity between questions and their ability of retrieving answers that can be interpreted by fuzzy logic [3, 16].

According to Zadeh, search engines should have the capability of answering questions formulated by a user and in order to achieve this objective, causality becomes an important factor that has to be modeled by fuzzy logic. There are two basic types of causality i) the so-called *forward causality* (“What are the effects caused by a concrete event?”) and ii) the *inverse causality* (“What actions have been provoked by a certain event?”). The first one is easier to deal with than the inverse causality because the involved action is usually known whereas for the inverse causality, there can be multiple factors that can have provoked an action, therefore it is more complex to be analyzed. Some attempts in order to deal with forward causality can be seen in [24, 15]. Also the causal relations are very related to how-questions that it is one of the most researched types of questions [19, 8].

Another challenge for this system is the use of fuzzy operators with respect to their use in classic information retrieval systems [10].

## 3 Castalia Architecture

Castalia project presents a different point of view for Q&A systems. Its objective is to handle several Q&A systems under a common interface and for this purpose a metasearch engine has been designed which will be used for testing fuzzy logic-based algorithms related to the different processes of the information retrieval.

All metasearch engines share the main components [21] but in this case new additional characteristics have to be added in order to carry out the goal of retrieving a unique answer for each question. The Castalia architecture can be seen in figure 1.

The system can be divided into two main parts, the support functions and the operative functions. The support modules are useful for accessing and managing external resources such as configuration files or electronic dictionaries. The operative modules perform the main functions of the metasearch engine. The access to the system is really simple, a web page with a text box where the user can submit his question in natural language.

As Castalia is being designed for research purposes

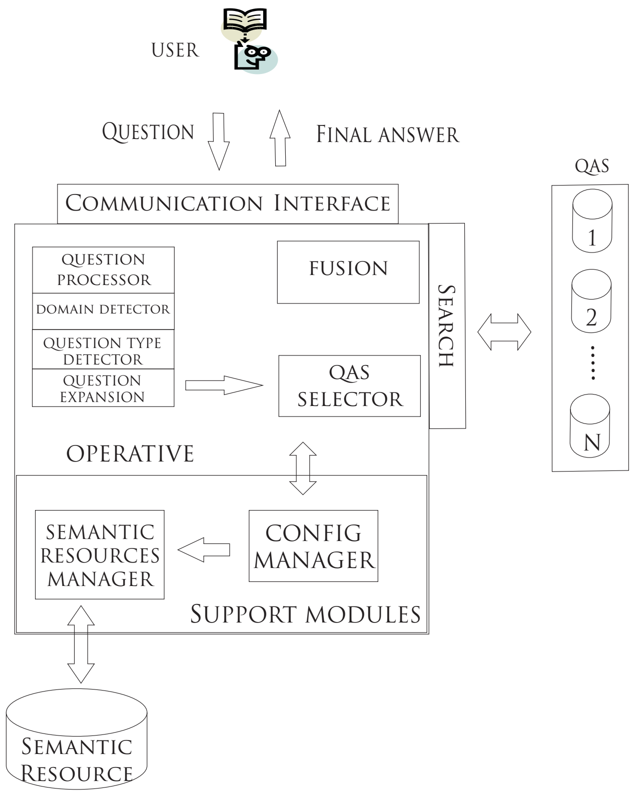

**Figure 1. Castalia Architecture**

different configuration profiles can be tested in order to achieve the optimal performance of the system just changing some parameters of the configuration files. Its design permits easily incorporating new algorithms for each phase of the search process and new Q&A systems, simply developing a new specific adapter. Even the appearence of new technologies is not a problem; Castalia is a Java-based system that can be deployed in any environment.

The usual data flow should be as follows (see Figure 2): the user submits a question in natural language. This question has to be analyzed by means of natural language techniques in order to extract its main characteristics. For example, it should be necessary to know what kind of question is in order to select the Q&A systems in which the query will be submitted. With the same goal it is necessary to identify the domain of the question: medicine, law, history, science, etc.

After this process the original query can be reformulated to generate new queries using external resources such as past queries, electronic dictionaries, domain ontologies, etc. In this step, those questions, which seem to be able to return results that could be interpreted by fuzzy logic, will be analyzed.

Once this step has finished, each question has to be submitted on one or several Q&A systems depending on different criteria as the domain of the Q&A system and the question or the kind of question previously identified. Typical metasearch engines collect all results and merge them in order to generate a ranked list containing the most relevant documents, on the contrary in this system, the results of the all Q&A systems will be processed in order to merge all answers into a unique answer. In this step the answers have to be represented by fuzzy logic in order to extract the necessary knowledge to generate a complete and consistent answer.

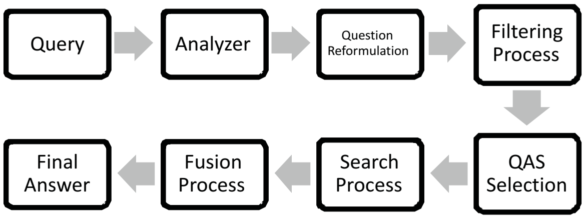

**Figure 2. Castalia Flow**

## 4 Operative Modules

The main operations of the system are performed in this subsystem: the detection of the question type and its domain, the selection of the suitable Q&A system, the question expansion, the search and the fusion.

### 4.1 Question Processor

Its responsibility is to interact with the other subsystems requesting and sending information in order the obtain the final answer.

Obviously this stage is more complicated because the question processing needs to be done depending on the different characteristics that are necessary in next steps such as the domain or the type of the question. Depending on the question, different architectures for this stage can be designed, some examples can be found in these works [12, 17].

#### 4.1.1 Question Type Detector

The functionality of this module is to detect the type of question submitted by the user. For this purpose several algorithms can be implemented. One of them is based on the analysis of the POS (Part-Of-Speech) tags of the first five terms of each question[22]. Analyzing hundreds of questions, a set of patterns has been created to determine whether a sentence is a question and what type of question. In this case GATE framework [7] has been the tool selected for detecting the different POS of the questions.

#### 4.1.2 Domain Detector

The functionality of this module is to discover the domain of each question. To discover the domain, several algorithms can be implemented and several lexicons can be used for each domain. In this module it is easy to add new lexicons and algorithms implementing a simple interface and configuring their parameters in the configuration file.

#### 4.1.3 Question expansion

A single question is not able to extract all the knowledge stored in a Q&A system, therefore alternative questions are necessary but it is a complicated task. It is not so easy to generate automatically for the question:

Why did Socrates die?

And it is very difficult to measure the relation degree between that question and this:

What killed Socrates?

This module is especially designed to study all these problems.

### 4.2 Searcher Selector

This module receives information from the previous modules: question type, domain and other characteristics in order to select the search engines that will perform the query. From these data a set of algorithms capable of selecting the most suitable Q&A systems and returning the result to Question Processor should be implemented.

### 4.3 Searcher

Its functionality is to search the answer requested by the user. For this purpose, the selector chooses which are the most suitable search engines and this subsystem submits the original question and the reformulated questions to each Q&A system.

Now there are several Q&A systems accessible such as START[3], NSIR[4], LAWGURU[5], FLASH-MED[6], WHOANEDIT[7], OCHEF[8], ANSWERS[9], YAHOO ANSWERS[10]. All of them belong to different domains: law, medicine, cooking, cuisine, general purpose, etc. and they can answer to different questions: who is..?, how to cook ..?, etc. The addition of new Q&A systems is also very easy.

### 4.4 Merger

The goal of this module is to merge the answer according to an algorithm enabled in the configuration file. Its objective is to obtain a unique final answer not a set of possible answers. This subsystem is fundamental because the answers could be fuzzified in order to check the consistency or the completeness and this is why it is so essential to detect the question type (causal, temporal, etc.). Hence the goal of this subsystem is to summarize a set of answers by means of logic-based algorithms.

Following the same principles of the architecture, an interface has been implemented that allows the incorporation of new fusion algorithms that will be configurable.

## 5 Support Modules

As can be seen the main tasks are processed by the above-mentioned modules but additional functionalities are necessary to support them.

### 5.1 Config Manager

The configuration module manages all the possible algorithms and resources for each step (fusion, search, domain detection, etc.) in order to design different search strategies. Since the configuration files are written in XML, any user can access and manipulate them using Xpath [5].

### 5.2 External Resources

Additional external information is necessary according to the final goal for Castalia. For instance, if the configuration for Castalia is focused on answering questions about bibliographical domains, maybe it would be necessary a lexicon containing names and surnames of famous people and their nicknames in order to reformulate or expand the original question. About a historical domain a list of important events, dates and people could be interesting.

Again for each new resource an interface is implemented that works as access point to its properties.

## 6 Conclusions and Future works

An architecture for managing Q&A systems jointly has been presented. It allows us to group several fuzzy concepts under the same platform. The development of this platform involves new paradigms such as the summarization of several answers from different sources using fuzzy logic, the selection of Q&A systems by using as main information the type of question that the user has submitted or the study of

[3] http://start.csail.mit.edu/

[4] http://tangra.si.umich.edu/clair/NSIR/html/nsir.cgi

[5] http://www.lawguru.com/cgi/bbs/user/search.cgi

[6] http://www.flash-med.com/Subjects.asp

[7] http://www.whonamedit.com/

[8] http://www.ochef.com/archive.htm

[9] www.answers.com

[10] http://answers.yahoo.com/

the capability for each Q&A system to retrieve information that can be modeled by fuzzy logic.

The platform is being developed now. Several Q&A systems have been connected and their answers have been preprocessed in order to have a common structure for the representation of the answers but it is necessary to have a larger number of different Q&A systems to select the most suitable for each question and to cover more domains. All modules developed up to now are configurable and they are prepared to configure different strategies that still have to be decided. Although some strategies for detecting question types have been already implemented further research is needed to develop different algorithms to exploit the system.

## 7 Acknowledgments

Partially supported by TIN2007-67494 F-META project, MEC-FEDER, Spain and PEIC09-0196-3018 SCAIWEB-2 excellence project, JCCM, Spain.

## References

[1] *Issues, Tasks and Program Structures to Roadmap Research in Question & Answering (Q&A)*, 2001.

[2] D. Ahn, S. Schockaert, M. D. Cock, and E. E. Kerre. Supporting temporal question answering: Strategies for offline data collection. In *5th International Workshop on Inference in Computational Semantics*, 2006.

[3] B. Bouchon-Meunier, M. Rifqi, and S. Bothorel. Towards general measures of comparison of objects. *Fuzzy Sets Systems*, 84:143–153, 1996.

[4] S. Calegari and E. Sanchez. A fuzzy ontology-approach to improve semantic information retrieval. In *Proceedings of the Third ISWC Workshop on Uncertainty Reasoning for the Semantic Web - URSW'07*, volume 327 of *CEUR Workshop Proceedings*, 2007.

[5] J. Clark and S. D. Rose. Xml path language (xpath) version 1.0 w3c recommendation. In *Technical Report REC-xpath-19991116. World Wide Web Consortium*, 1999.

[6] F. Crestani and G. Pasi. *Soft computing in information retrieval: Techniques and applications*. Physica Verlag, 2000.

[7] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. Gate: A framework and graphical development environment for robust nlp tools and applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics*, 2002.

[8] R. Girju and D. Moldovan. Mining answers for causation questions. In *AAAI Symposium on Mining Answers from Texts and Knowledge Bases*, 2002.

[9] B. F. Green Jr., A. K. Wolf, C. Chomsky, and K. Laughery. Baseball: an automatic question answerer. In *Computers and thought*, pages 207–216, Cambridge, MA, USA, 1995.

[10] E. Herrera-Viedma, J. L. Gijón, S. Alonso, J. Vílchez, C. García, L. Villén, and A. G. López-Herrera. Applying aggregation operators for information access systems: An application in digital libraries. *International Journal of Intelligent Systems*, 23:1235–1250, 2008.

[11] W. Meng, C. Yu, and K.-L. Liu. Building efficient and effective metasearch engines. *ACM Computing Surveys*, 34:48–89, 2002.

[12] S. Narayanan and S. Harabagiu. Question answering based on semantic structures. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, page 693, Morristown, NJ, USA, 2004.

[13] M. Nikravesh, V. Loia, and B. Azvine. Fuzzy logic and the internet (flint): Internet, world wide web, and search engines. *Soft Computing*, 6:287–299, 2002.

[14] J. A. Olivas. Fuzzy sets and web meta-search engines and soft computing. In *Fuzzy Sets and Their Extensions: Representation, Aggregation and Models: Intelligent Systems from Decision Making to Data Mining, Web Intelligence and Computer Vision*, pages 538–554, 2007.

[15] C. Puente and J. A. Olivas. Analysis, detection and classification of certain conditional sentences in text documents. In *Proceedings of the 12th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 1097–1104, 2008.

[16] M. Rifqi, V. Berger, and B. Bouchon-Meunier. Discrimination power of measures of comparison. *Fuzzy Sets Systems*, 110:189–196, 2000.

[17] E. Saquete, P. Martínez-Barco, R. Muñoz, and J. L. Vicedo. Splitting complex temporal questions for question answering systems. In *ACL '04: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, pages 566–573, Morristown, NJ, USA, 2004.

[18] S. Schockaert, D. Ahn, M. D. Cock, and E. E. Kerre. Question answering with imperfect temporal information. *In Proceedings of the 7th International Conference on Flexible Query Answering System*, pages 647–658, 2006.

[19] R. Schwitter, F. Rinaldi, and S. Clematide. The importance of how-questions in technical domains. In *Proceedings of the Question- Answering workshop of TALN 04*, 2004.

[20] J. Serrano-Guerrero, F. P. Romero, and J. A. Olivas. Filtering short queries by means of fuzzy semantic-lexical relations for meta-searchers using wordnet. In *Web Intelligence and Intelligent Agent Technology, IEEE/WIC/ACM International Conference on*, volume 3, pages 269–272, Los Alamitos, CA, USA, 2008.

[21] J. Serrano-Guerrero, F. P. Romero, J. A. Olivas, J. De La Mata, and A. Soto. Budi: Architecture for fuzzy search in documental repositories. *Mathware and Soft Computing*, 16:71–85, 2009.

[22] L. Shrestha and K. McKeown. Detection of question-answer pairs in email conversations. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, page 889, Morristown, NJ, USA, 2004.

[23] A. Soto, J. A. Olivas, and M. E. Prieto. Using generalized constraints and protoforms to deal with adverbs. In *EUSFLAT Conf. (2)*, pages 119–126, 2007.

[24] E. Trillas. Lógica borrosa y narrativa: un párrafo de vilamatas. In *XII Congreso Español sobre Tecnologías y Lógica Fuzzy ESTYLF'04*, pages 23–34, 2004.