QALL-ME needs AIR: a portability study

Constantin Orăsan, Iustin Dornescu and Natalia Ponomareva
Research Group in Computational Linguistics
University of Wolverhampton, UK
C.Orasan, I.Dornescu2, Nata.Ponomareva@wlv.ac.uk

Abstract

Currently access to institutional repositories is gained using dedicated web interfaces where users can enter keywords in an attempt to express their needs. In many cases this approach is rather cumbersome for users who are required to learn a syntax specific to that particular interface. To address this problem, we propose to adapt the QALL-ME framework, a reusable framework for fast development of question answering systems, in order to allow users to access information using natural language questions. This paper describes how the web services part of the QALL-ME framework had to be adapted in order to give access to information gathered from unstructured web pages by the AIR project.

Keywords

QALL-ME framework, web services, question answering, textual entailment

1 Introduction

Currently access to institutional repositories is gained using dedicated web interfaces where users can enter keywords. In many cases this approach is rather cumbersome for users who are required to learn a syntax specific to that particular interface. A solution to this problem is offered by question answering (QA), a field in computational linguistics, which develops systems that take a natural language question and provide the exact answer to it. This paper presents how the QALL-ME framework[1], a reusable framework for fast development of question answering systems, was adapted in order to allow users to access information stored in institutional repositories using natural language questions.

The AIR project [10] developed a system that extracts information about scientific publications from unstructured documents and stores this information in a database. The QALL-ME project [12] has developed a framework for implementing question answering systems for restricted domains. The first implementation of this framework was for the domain of tourism, but it is not in any particular way bound to this domain. For this reason, the QALL-ME framework can offer the ideal and most natural way of accessing the information extracted in the AIR project.

The remainder of the paper is structured as follows: Section 2 presents some background information about the QALL-ME and AIR projects. The domain in which the system is expected to run is presented in Section 3, followed by a description of the framework and how it was adapted to the new domain in Section 4. The paper finishes with conclusions.

2 Background information

2.1 The QALL-ME project

QALL-ME (Question Answering Learning technologies in a multiLingual and Multimodal Environment) is an EU-funded project with the objective of developing a shared infrastructure for multilingual and multimodal open domain Question Answering.[2] It allows users to express their information needs in the form of multilingual natural language questions using mobile phones and returns a list of ranked specific answers rather than the whole web pages. In the first phase, the tourism domain is highlighted as the domain in which the operates.

Language variability, one of the main difficulties of dealing with natural language, is addressed in QALL-ME by reformulating it as a textual entailment recognition problem. In textual entailment a text (T) is said to entail a hypothesis (H), if the meaning of H can be derived from the meaning of T. To this end, each question is treated as the text and the hypothesis is a procedure to answer the question [6]. This concept is embedded in the QALL-ME Framework [12], one of the main outputs of the project. The purpose of this framework is to provide an architecture skeleton for QA systems that extract answers from structured data sources. This framework is exploited in this paper to provide an access to data collected by the AIR project.

2.2 The AIR project

Manual population of institutional repositories with citation data is an extremely time- and resource-consuming process, and usually acts as a bottleneck on the fast growth and update of large repositories. The aim of the AIR project [10] was to develop a semi-automatic approach for archiving institutional repositories. To achieve this, it automatically

[1] The QALL-ME framework is available as an open source project at `http://qallme.sourceforge.net/`

[2] More information about the QALL-ME project can be found at `http://qallme.fbk.eu`

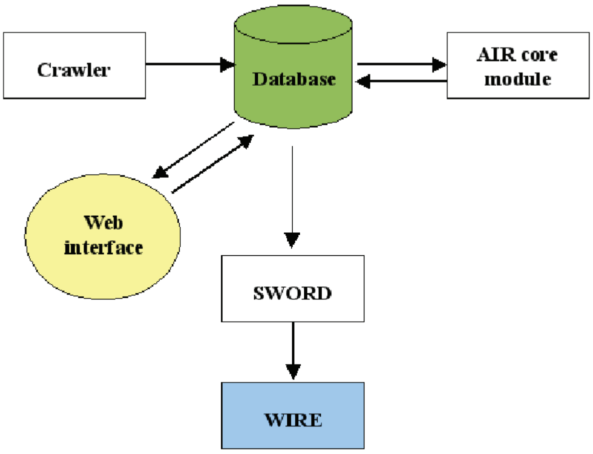


Fig. 1: *Overview of the AIR architecture*

discovers and extracts bibliographical data from web sites, and then interacts with users, authors or librarians, who verify and correct extracted data. The components of the AIR system are:

1. A **web crawler** which populates the database with all web pages belonging to the domain under consideration.

2. the **AIR core module** which processes unstructured data (web pages) in order to extract bibliographical references and automatically annotate them with Dublin Core Metadata tags.

3. **Web interfaces** developed for user interaction and data verification. This step was introduced to ensure the reliability of information transferred to the digital repository.

4. Data deposit is accomplished automatically using SWORD[3] protocol.

An overview of the system is presented in Figure 1.

In this section only the information extraction component is briefly presented. The other components are not relevant for the current research; more details about them can be found in [10].

Automatic extraction of bibliographical metadata from unstructured web pages is achieved in three consecutive steps using a machine learning approach:

1. A page classifier extracts web pages containing bibliographies from the whole amount of data provided by the crawler. The classifier exploits the structure of HTML format such as metadata (<keywords>, <title>, <description>) and headers (<h1>, <h2>, etc.) in order to give different weights to distinct data sources. As a machine learning method, we used and compared several methods contained in WEKA [17].

2. A record classifier selects bibliographical entries from all document records using Conditional Random Fields (CRF) [4]. As HTML-format provides some structural elements like tags, we incorporated them into the classifier for revealing enumerations or lists of equivalent records.

3. An information extraction (IE) module identifies 5 types of bibliographical metadata from Dublin Core Metadata Element Set: author, title, date, publisher and citation. As bibliographical reference represents a logical consequence of metadata tags, the use of CRF is the most appropriate.

Given that all three modules focus on similar problems we used very similar features while constructing corresponding classifiers. These features are:

- **Named Entities**, such as PERSONS, LOCATIONS, ORGANIZATIONS and DATES identified using ANNIE[4], a named entity recogniser distributed together with GATE[5] framework [2].

- **Staff names**: A list of all university members was collected and used to annotate the input text.

- **Sherpa/Romeo publishers and journals** the list of publishers and journals stored in the Sherpa/Romeo database was retrieved and used to annotate the input text.

- **Presence of year**: This feature indicates whether an element contains a year.

- **Publication triggers**: We built different lists of triggers that can indicate different types of information about a publication. Examples of such triggers are header indicators (the most frequent words that occur in <h> tags of files containing publications), publication triggers (words appearing in a bibliographical entry itself), citation triggers (words appeared in citation of an entry).

- **Orthographic features**, which capture capitalisation, digits, punctuation marks, etc. This feature was only used for implementation of the IE module.

- **Parts of speech (POS)**. Preliminary experiments revealed many cases when automatically annotated bibliographical fields ended with articles, conjunctions or prepositions. In order to correct this situation, we incorporated parts of speech information into the IE module.

All three modules of the IE component were evaluated on manually annotated data using cross-validation [10]. The page classifier obtained an f-score of 0.882 when using JRIP [1]. The record classifier obtains the best results (f-score of 0.919) with a Markov order 1. The information extraction module can recognise the author, date, title and citation with high accuracy, but perform rather poorly on identification of publisher. Detailed evaluation results can be found in [10].

[3] http://www.ukoln.ac.uk/repositories/digirep/index/SWORD_guide

[4] http://gate.ac.uk/ie/annie.html

[5] http://gate.ac.uk/

3 Domain description and modelling

An investigation of the domain of bibliographical references and the data extracted by the AIR project was carried out in order to define which questions can be answered by the system and to model the domain using an ontology. This section describes the domain and ontology used to represent this domain.

3.1 The domain

The data stored by the AIR information extraction module in the database contains information about the author(s) of a publication, its title, year of publication, publisher and the rest of the information lumped together as "citation". Using the annotation modules presented in Section 4.2 it is possible to extract from the citation the name of the journal or conference proceeding that published the article. Using automatic term extraction techniques or external databases it is possible to automatically assign keywords to each article. On the basis of this information, the following types of questions are foreseen to be answered by the system:

1. Questions about the name of an author who published in a year, in journal/conference and/or on a topic (e.g. *Which authors had a publication in <JOURNAL> in <YEAR>?*)
2. Questions about the year when an author published in a journal/conference, on a topic or with other authors (e.g. *What year did <AUTHOR> have a paper in <JOURNAL>?*)
3. Questions about the title of a publication by an author, in a journal/conference, on a topic, in a certain year and/or with other authors (e.g. *What papers about <KEYWORDS> did <AUTHOR> publish in <YEAR>?*)
4. Questions about the title of a conference/journal where an author published on a topic, in a certain year and/or with other authors (e.g. *Who published <AUTHOR> in <YEAR>?*)
5. Questions about the topic of an article published by an author in a journal/conference, in a certain year and/or with other authors (e.g. *What are the topics of the papers published in <YEAR> by <AUTHOR> and <AUTHOR>?*)

Each question can have one or several constraints, but none expects the user to specify the title of the publication. This is due to the fact that scientific articles tend to have long titles, which are unlikely to be remembered correctly and completely by a user. For this reason, even with the fuzzy matching implemented in our annotators, it is unlikely that the system can correctly guess what title a user is referring to. On the basis of the five types of questions identified above, 36 types of questions were proposed to be answered by the system.

3.2 The ontology

The purpose of the ontology is to provide a conceptualised description of the selected domain and to act as a link between different components of the system and different languages. The ontology developed in QALL-ME for the domain of tourism is described in [9]. Given that the scope of the AIR project consists of academic citations, we could not use this ontology and instead we had to find an ontology which:

- uses standard metadata terminologies such as Dublin Core (dc and dcterms);
- supports the entry types used by the open BibTeX reference management software or other similar schemes;
- allows arbitrary keyword indexing schemes; and
- uses dereferenceable URIs for interoperability with other systems (faceted browsing, semantic web mash-ups)

We decided to use BibTeX as it is very popular in the academic community and it is supported by many citation management systems. Moreover, the format in which AIR stores data can be easily mapped into the BibTeX format. Fields which are not explicitly identified by AIR, such as the name of the proceedings, can be easily extracted using the annotators presented in Section 4.2. In addition, by using this approach, we are not limited to using only the output of the AIR project and we can apply our QA interface to a large number of sources.

The advantage of using BibTeX as the format of the input data is that there are several ontologies that can be used (e.g. the MIT bibtex ontology[6], bibo[7], SWRC[8]. The differences between them are the vocabularies used and details such as author list representation and event representation. We chose to use a subset of the SWRC ontology [14], an ontology for modelling entities of research communities such as persons, organisations, publications (bibliographic metadata) and relationships amongst them. The main entities involved are: persons (authors and editors), organisations (publishers, research institutes, universities), publications (articles, conference papers, theses, book chapters) and collections (proceedings, journals, books, series). A relevant subset of the Dublin Core metadata terminology is used to describe the properties of the bibliographic entries. An example of a conference paper in the TURTLE syntax defined by our ontology can be seen in Listing 1.

[6] http://zeitkunst.org/bibtex/0.1/

[7] Bibliographic Ontology Specification http://bibliontology.com/

[8] Semantic Web for Research Communities http://ontoware.org/projects/swrc/

Listing 1: *Example of conference paper presented in TURTLE syntax*

```
qa2:Mitkov1998
rdf:type swrc:InProceedings;
dc:title ”Robust pronoun resolution with
     limited knowledge”;
terms:issued ”1998”;
swrc:pages ”869−875”;
terms:isPartOf qa2:conf/acl/2008;
dc:creator qa2:Mitkov_R_.

qa2:conf/acl/2008
rdf:type swrc:Proceedings;
dc:title ”Proceedings of the 18th
    International Conference on
    Computational Linguistics (COLING
    ’98)/ACL’98 Conference”;
swrc:address ”Montreal, Canada”.
```

The SWRC terminology is also used by the DBLP[9] (Digital Bibliography & Library Project) computer science bibliography website. This makes it easy to augment the data collected by the AIR project with bibliographic information from other sources. This can be further extended by employing protocols such as Open Archives Initiative (OAI) Metadata Protocol Handler which is an interchange format that facilitates metadata harvesting from electronic repositories or the PRISM[10] protocol. Whilst AIR does not provide us with details such as affiliation relations, this information could be added from such sources, enabling the system to answer questions such as *”Scientists working in which German universities have published papers about Question Answering in 2008?”*.

Using existing software, the data collected by the AIR project in BibTeX format was converted to the RDF format defined by our ontology. For data access the SPARQL query language was used.

3.3 Representation of terms

As seen in Section 3.1, a large number of questions that can be asked are restricted by topics. These topics are normally represented using terms and therefore it was necessary to find a convenient way to represent terminologies and relationships between them.

Investigation of existing resources revealed that the `skos` (Simple Knowledge Organisation System) ontology [11] is appropriate as it provides a model for expressing the basic structure and content of concept schemes. A *concept scheme* is defined in the `skos` ontology as “a set of concepts, optionally including statements about semantic relationships between those concepts.” (e.g. thesauri, classification schemes, terminologies, glossaries, etc.) The `skos` ontology is useful for our purposes as it encodes relations such as `skos:broader`, `skos:narrower`, `skos:broaderTransitive` and `skos:narrowerTransitive` and allows the asking of questions such as: *”What did Constantin Orasan publish about summarization”?* and the retrieval of papers which were tagged with *multi-document summarization.* A part of the terminology corresponding to computational linguistics can be found in Figure 2.

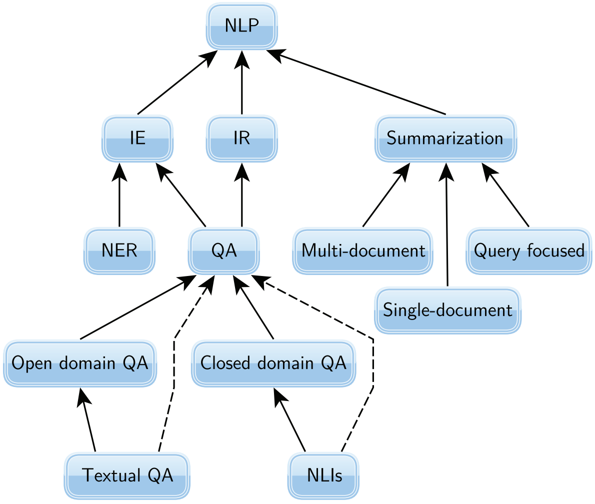


Fig. 2: *Test indexing terminology (full lines represent the* `skos:broader` *relation while the dotted lines represent the* `skos:broaderTransitive` *relation)*

4 The QALL-ME framework and its adaptation to a new domain

The QALL-ME framework is based on a Service Oriented Architecture (SOA) which, for the domain of tourism, is realised using the following web services:

1. Context providers: are used to anchor questions in space and time in this way enabling answers to temporally and spatially restricted questions
2. Annotators: Currently three types of annotators are available:
    - named entity annotators which identify names of cinemas, movies, persons, etc.
    - term annotators which identify hotel facilities, movie genres and other domain-specific terminology
    - temporal annotators that are used to recognise and normalise temporal expressions in user questions
3. Entailment engine: is used to overcome the problem of user question variability and determine whether a user question entails a retrieval procedure associated with predefined question patterns.
4. Query generator: which relies on an entailment engine to generate a query that can be used to extract the answer to a question from a database.

[9] `http://dblp.uni-trier.de/`

[10] `http://www.prismstandard.org/about/`

[11] (http://www.w3.org/TR/2005/WD-swbp-skos-core-guide-20050510/)

For the tourism demonstrator the output of this web service is a SPARQL query.

5. Answer pool: retrieves the answers from a database. In the case of the tourism demonstrator the answers are extracted from RDF encoded data using SPARQL queries.

Given that the QALL-ME framework was implemented using a modular approach, in order to adapt the system to a new domain all that was required was to re-implement or adapt some of the existing web services. This section describes each of the web services in more detail emphasising the changes that had to be made in order to adapt the system to the new domain.

4.1 Context provider web service

Investigation of the bibliographical domain revealed that the role of this web service is rather limited as the spatial information is not used in questions. For this reason, this web service was reduced to returning the current date and time in order to be able to answer questions with temporal restrictions such as:

What papers were published in 2009?.

What papers were published ***this year****?.*

The context information is used by the temporal annotator (described in Section 4.2.3) to normalise the *this year* expression to the TIMEX2 standard.

4.2 Annotators

The QALL-ME framework provides the possibility of annotating the input sentences with named entities, terms specific to the domain and temporal expressions. This section presents the annotator services implemented for the bibliographical domain.

4.2.1 Named entity and term annotators

Unlike standard Named Entity Recognition that are able to determine unknown and new entities, in restricted domains the entities are known and the annotation task is reduced to a database look-up that needs to deal with spelling mistakes, inaccurate entity references and partial matches. The ability to deal with noisy input becomes even more important for the domain of tourism where users have the possibility of asking questions using speech which means that the system needs to deal with automatic speech recognition errors.

For the domain of tourism we had to recognise named entities such as names of hotels, movies and persons, as well as terms which are multi-word expressions related to the domain such as genre of a movie (e.g. action movie) and site facilities (e.g. disabled access). Both types of expressions are identified using the same greedy algorithm that annotates expressions from a gazetteer, based on a character-based similarity distance between the tokens and an adapted TFIDF score.

The main difference between our method and other approaches, such as [5] and TagLink [13], is that it distinguishes between high probability similar tokens and tokens that are most probably distinct. Using a character-based similarity threshold, we compute a partial one to one matching and evaluate the negative impact of the unmatched tokens. The more words that are matched, the greater the confidence that the two strings represent the same entity, while a great number of mismatched words means that the two strings represent different entities. While matching tokens a certain amount of spelling variations and mistakes are allowed by using the character-based similarity measure proposed by Jaro-Winkler [16].

For the bibliographical domain the same approach was employed, but different types of entities had to be annotated. This was achieved by training the algorithm with different gazetteers. Given the nature of the domain, a large number of ambiguities were noticed. These ambiguities are discussed in the next section.

4.2.2 Ambiguities at the level of entity annotation

One of the main challenges we had to address when we ported the annotators to the new domain was that the names of authors and conferences can be expressed in several ways and that some entities can have several types. The former problem is referred to as **instance ambiguity** (IA) – entities having the same type and the same name, and the latter **type ambiguity** (TA) – entities of different types which have the same name (common for acronyms). For example in *What papers has Ruslan Mitkov published in computational linguistics?* the expression *computational linguistics* can be both a named entity and refer to the *Journal of Computational Linguistics*, or can be a term and refer to the field of *computational linguistics*. This represents a type ambiguity. In the same question, it is possible to refer to *Ruslan Mitkov* using *R. Mitkov* or just *Mitkov* which exemplifies an instance ambiguity.

The Entity Annotator marks spans of text from the question with the type and ID of the entities they refer to. It retrieves a list of all the entities of a given type (e.g. [*swrc:Person*]) from the RDF data along with their known aliases and creates an index which will be used to identify and rank candidate matches in questions. The Entity Annotator aggregates several distinct annotators, one for each type (e.g. persons, publishers, conferences) and applies specialised semantic rules when indexing the lists extracted from the RDF data (e.g. derive the alias "Dornescu I." from "Dornescu Iustin", or "ACL08" from "ACL 2008"). Employing such semantic rules off-line, during data pre-processing is computationally motivated. At query time, the annotator ranks all the known alternative names (synonymy) and is also able to select an ambiguous list of entities which are equally ranked (polysemy). The latter is an extension of the initial QALL-ME web services, which allowed any span of the question to refer to at most one entity.

Since the QALL-ME web services definition does not allow annotated spans to overlap, when ranking the candidates, the annotator will prefer longer spans and fewer distinct IDs. This constrains both types of ambiguity. For example when the user inputs the following question:

> *What did I. Dornescu publish in CLEF 2008?*

the following candidates are considered:

```
 1 qa2:dornescu/iustin    "Dornescu Iustin"
 2 qa2:dornescu/iustin  * "Dornescu I."
 3 qa2:dornescu/iulian    "Dornescu Iulian"
 4 qa2:dornescu/iulian  * "Dornescu I."
 5 qa2:i_/dornescu      * "I. Dornescu"
 6 qa2:dornescu           "Dornescu"
 7 qa2:domnescu/I.        "Domnescu I."
 8 qa2:conf/clef/2008   * "CLEF 2008"
 9 qa2:conf/clef        * "CLEF"
10 qa2:lcns/clef2008    * "CLEF 2008"
11 qa2:ev/wk/clef/08    * "CLEF 2008"
```

has three ambiguous author matches (two of which could be duplicates), and three ambiguous acronym matches. Candidate 6 is discarded because its span overlaps a larger one which has priority (more matching words means a greater confidence); candidate 7 has a lower confidence score due to the edit distance; while candidates 1 and 3 are aliases of the highest ranked candidates. The advantage when using this scheme is that the system can inform the user that "I. Dornescu" is ambiguous and can suggest which are the alternatives. "CLEF 2008" can refer to: an event (an workshop), a series of events, or two published proceedings (the electronic Working Notes and the LCNS volume).

In contrast to the annotators used for the domain of tourism, due to the large number of ambiguities that can be present in the questions, the annotators used here return all the possible annotations of a question, leaving the rest of the components to select the correct interpretations.

4.2.3 Temporal annotator

Investigation of the domain of tourism, in which the QALL-ME framework was initially developed, revealed that a large number of questions contain temporal constraints. For this reason, the framework provides the possibility of using a temporal annotator that identifies temporal expressions and normalises them using the TIMEX2 standard [3].

The temporal tagger used for the domain of tourism follows the design and methodology of the temporal tagger described in [11] that is capable of identifying both self-contained temporal expressions(TEs) and indexical/under-specified TEs. The annotator described in [11] is rule-based and tackles more cases than necessary, making it too slow for our purposes. For this reason, a simplified temporal annotator was implemented [15]. Evaluation of this simplified temporal annotator revealed that it performs with high accuracy, most of the errors being due to the reduced number of rules implemented to increase its speed and ambiguities specific to the domain of tourism.

In contrast, questions about publications features very few temporal expressions, most of them being references to years (e.g. *Which journals/conferences published [AUTHOR] in [YEAR]?*). Theoretically it is possible to use more precise temporal expressions which specify both the month and the year, or even the day, but there are very few bibliographical databases which contain enough information to allow retrieval of articles based on the precise date when they were published. In light of this, we decided to use the temporal annotator implemented for the domain of tourism without any change, knowing that it can handle without a problem both references to years and specific dates. In addition, it can deal with indexical references such as *this year* without a problem.

The output of the temporal annotator is passed to the TIMEX2SPARQL web service which converts the TIMEX2 expressions to SPARQL snippets that are used to extract the answer to questions. This web service was used directly from the QALL-ME framework.

4.3 Entailment engine

The most common way to answer questions in restricted domains is to take a natural language question and transform it to a standard query language such as SQL. Often this requires performing deep linguistic analysis and reconstructing the logical form of the question. Despite the extensive manual work that goes into such a method, this approach fails quite often due to the language variability which allows the same question to be expressed in numerous ways. This problem is addressed in the QALL-ME project by using an entailment module that determines whether different expressions entail the same meaning and thus can share the same retrieval procedure [6]. The retrieval procedure is a SPARQL[12] template that is instantiated by the query generator (Section 4.4).

The English prototype that works in the domain of tourism uses an entailment engine which relies on domain ontology and its alignment to WordNet [9] to calculate the similarity between two questions and determine whether there is an entailment relation between them. Before this similarity score is calculated, as a pre-processing step, the expected answer types of the two questions and the types of entities appearing in the questions are compared and if they are not compatible the entailment relation is rejected [7].

For the entailment engine used in the bibliographical domain, we could not use the existing entailment engine as we could not easily incorporate the new domain ontology and we do not have its alignment to WordNet. Instead, we adapted the expected answer type module to recognise the five types of questions presented in Section 3.1 and use the similarity metric distributed together with the framework which is language independent. This similarity metric combines Levenshtein distance, cosine similarity, Euclidean distance and Monge Elkan distance. Despite this rather simple approach, the results are good.

[12] `http://www.w3.org/2001/sw/DataAccess/`

4.4 Query generator

The role of the Query generator web service is to produce a valid SPARQL query which can be used to answer a given question. To achieve this, it has a pre-prepared set of question patterns together with their retrieval procedure. The query generator relies on the entailment engine to determine which question pattern is entailed by a user question and in this way determine the retrieval procedure. The retrieval procedure is a SPARQL template which contains slots that are filled in using entities from the question.

The query generator used for the domain of tourism had to be changed in order to be used for the new domain. The first change was to produce question patterns and their SPARQL templates for the new domain. Ou et. al. [8] show how these can be automatically produced starting from the ontology, but the results they report are lower than for the manually produced set. In light of this, the question patterns and their SPARQL templates were manually produced on the basis of the types presented in Section 3.1.

The second change introduced was to allow the query generator to deal with ambiguous entities. The original query generator expects that each entity has exactly one interpretation. This is not the case in the bibliographical domain and the query generator was modified to gather all the possible interpretations when calling the entailment engine. For example, in

> *What did I. Dornescu publish in CLEF 2008?*

CLEF 2008 can be interpreted both as name of proceedings and as an event. Given that there is no question pattern referring to an event, the entailment engine is used to rule out the interpretation where *CLEF 2008* is an event.

The SPARQL query generated in this web service is used in the next step to retrieve the actual answer.

4.5 Answer pool

The role of the Answer Pool web service is to take the SPARQL query generated by the query generator and retrieve the results from an RDF database. The answer pool service is domain independent and was used with almost no changes. For cases where entities are ambiguous (both type and instance ambiguities) and the question can be interpreted in several ways, several SPARQLs are generated and therefore several answer sets are retrieved. This represents an improvement over the service used in the domain of tourism.

The answer pool also plays a role in dealing with ambiguities. Some SPARQL queries retrieve no results due to the interpretations of some entities (e.g. in the case of *What did I. Dornescu publish in CLEF 2008?*, *I. Dornescu* can be interpreted as *Iulian Dornescu* who is not an author in our database). In this case the ambiguity is not shown to the user.

For cases where several interpretations of the question yield answers, a presentation module is used to show the results in a user-friendly manner. This is presented in the next section.

4.6 Presentation module

The presentation module is domain dependent and for this reason is not part of the framework. For the domain of tourism, the output of the system is textual answers, speech, maps, images and videos and different presentation modules were used depending on where the results were displayed (i.e. computer screen or mobile phone). This are not appropriate for the bibliographical domain, where for presenting the results to the user, we currently use the Citeline, a tool which turns a publication list in BibTeX format into a visual exhibit.[13] To be able to use Citeline the RDF data retrieved by the Answer Pool web service is converted back to BibTeX format. A screenshot of the presentation module is presented in Figure 3. No language processing is performed at this stage and ambiguities present in the input question are dealt with using faceted browsing.

5 Conclusions

This paper presented the adaptation of the QALL-ME Framework to the bibliographical domain. This process required to create a domain ontology and adapt the web services that constitute the framework.

To model the bibliographical domain, a subset of the SWRC ontology [14], an ontology for modelling entities of research communities such as persons, organisations, publications (bibliographic metadata) and relationships amongst them was used. The terminology specific to the domain was encoded using the `skos` ontology.

The QALL-ME Framework is based on a Service Oriented Architecture (SOA) and realised using web services. Each of the web services were described with emphasise on the changes necessary due to the new domain. One of the main problems that had to be faced was the large number of ambiguities (both instance and type ambiguities) that can be present in a user question. As a result, the annotation web services had to be changed to allow multiple annotations for a text span. This in turn, determined changes in the query generator and answer pool web services, in order to allow them to deal with multiple interpretations for a question.

Currently no formal evaluation of the system was carried out. In the future, we plan to collect questions from users and perform an on-field evaluation in order to be able to assess the performance of the system.

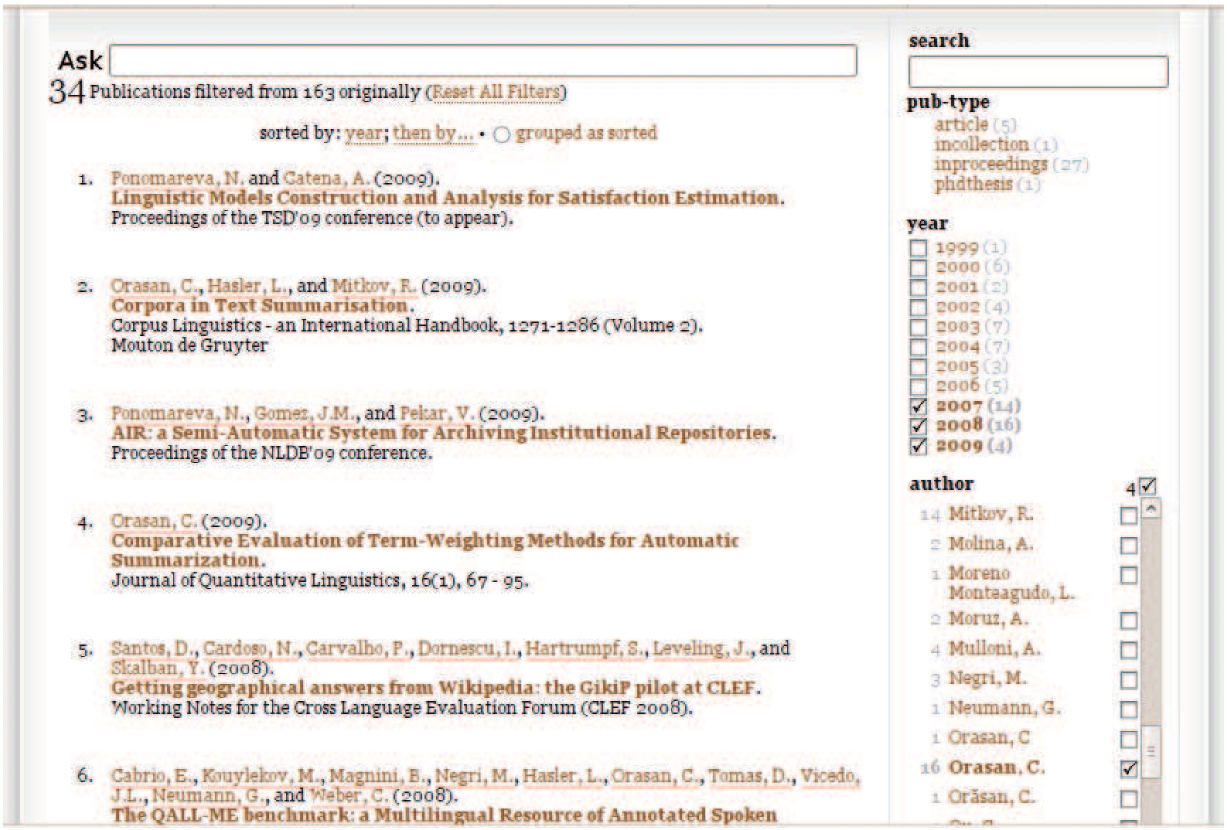
Acknowledgements

This work is supported by the EU-funded project QALL-ME (FP6 IST-033860).

References

[1] W. W. Cohen. Fast effective rule induction. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 115–123, 1995.

[13] `http://citeline.mit.edu/`



34 Publications filtered from 163 originally (Reset All Filters)

sorted by: year; then by... • ○ grouped as sorted

1. Ponomareva, N. and Catena, A. (2009).
Linguistic Models Construction and Analysis for Satisfaction Estimation.
Proceedings of the TSD'09 conference (to appear).

2. Orasan, C., Hasler, L., and Mitkov, R. (2009).
Corpora in Text Summarisation.
Corpus Linguistics - an International Handbook, 1271-1286 (Volume 2).
Mouton de Gruyter

3. Ponomareva, N., Gomez, J.M., and Pekar, V. (2009).
AIR: a Semi-Automatic System for Archiving Institutional Repositories.
Proceedings of the NLDB'09 conference.

4. Orasan, C. (2009).
Comparative Evaluation of Term-Weighting Methods for Automatic Summarization.
Journal of Quantitative Linguistics, 16(1), 67 - 95.

5. Santos, D., Cardoso, N., Carvalho, P., Dornescu, I., Hartrumpf, S., Leveling, J., and Skalban, Y. (2008).
Getting geographical answers from Wikipedia: the GikiP pilot at CLEF.
Working Notes for the Cross Language Evaluation Forum (CLEF 2008).

6. Cabrio, E., Kouylekov, M., Magnini, B., Negri, M., Hasler, L., Orasan, C., Tomas, D., Vicedo, J.L., Neumann, G., and Weber, C. (2008).
The QALL-ME benchmark: a Multilingual Resource of Annotated Spoken

pub-type
article (5)
incollection (1)
inproceedings (27)
phdthesis (1)

year
□ 1999 (1)
□ 2000 (6)
□ 2001 (2)
□ 2002 (4)
□ 2003 (7)
□ 2004 (7)
□ 2005 (3)
□ 2006 (5)
☑ 2007 (14)
☑ 2008 (16)
☑ 2009 (4)

author 4☑
14 Mitkov, R. □
2 Molina, A. □
1 Moreno Monteagudo, L. □
2 Moruz, A. □
4 Mulloni, A. □
3 Negri, M. □
1 Neumann, G. □
1 Orasan, C □
16 Orasan, C. ☑
1 Orăsan, C. □

Fig. 3: *Faceted browsing of results using powered by MIT Citeline, Babel and Exhibit*

[2] 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 ACL02*, 2002.

[3] L. Ferro, L. Gerber, I. Mani, B. Sundheim, and W. G. TIDES 2005 Standard for the Annotation of Temporal Expressions. Technical report, MITRE, April 2005.

[4] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[5] A. E. Monge and C. Elkan. The field matching problem: Algorithms and applications. In *KDD*, pages 267–270, 1996.

[6] M. Negri, B. Magnini, and M. O. Kouylekov. Detecting expected answer relations through textual entailment. In *9th International Conference on Intelligent Text Processing and Computational Linguistics*, pages 532–543, Heidelberg, Germany, 2008. Springer.

[7] S. Ou, D. Mekhaldi, and C. Orăsan. An ontology-based question answering method with the use of textual entailment. In *Proceedings of NLPKE09*, 2009.

[8] S. Ou, C. Orăsan, D. Mekhaldi, and L. Hasler. Automatic question pattern generation for ontology-based question answering. In *Proceedings of the 21st International Florida Artificial Intelligence Research Society Conference (FLAIRS2008)*, pages 183 – 188. Menlo Park, CA: AAAI Press, 2008.

[9] S. Ou, V. Pekar, C. Orăsan, C. Spurk, and M. Negri. Development and alignment of a domain-specific ontology for question answering. In European Language Resources Association (ELRA), editor, *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, May 28 – 30 2008.

[10] N. Ponomareva, J. M. Gomez, and V. Pekar. Air: a semi-automatic system for archiving institutional repositories. In *Proceedings of the 14th International Conference on Applications of Natural Language to Information Systems (NLDB 2009)*, Saarbrucken, Germany, June 24-26 2009.

[11] G. Puşcaşu. A framework for temporal resolution. In *Proceedings of the 4th Conference on Language Resources and Evaluation (LREC 2004)*, Lisbon, Portugal, May, 26-28 2004.

[12] B. Sacaleanu, C. Orasan, C. Spurk, S. Ou, O. Ferrandez, M. Kouylekov, and M. Negri. Entailment-based question answering for structured data. In *Coling 2008: Companion volume: Posters and Demonstrations*, pages 29 – 32, Manchester, UK, August 2008.

[13] A. Salhi and H. Camacho. A string metric based on a one-to-one greedy matching algorithm. *Research in Computer Science*, number 19:171–182, 2006.

[14] Y. Sure, S. Bloehdorn, P. Haase, J. Hartmann, and D. Oberle. The swrc ontology - semantic web for research communities. In C. Bento, A. Cardoso, and G. Dias, editors, *Proceedings of the 12th Portuguese Conference on Artificial Intelligence - Progress in Artificial Intelligence (EPIA 2005)*, volume 3803 of *LNCS*, pages 218 – 231, Covilha, Portugal, DEC 2005. Springer.

[15] A. Varga, G. Puşcaşu, and C. Orăsan. Identification of temporal expressions in the domain of tourism. In *Knowledge Engineering: Principles and Techniques*, volume 1, pages 29 – 32, Cluj-Napoca, Romania, July 2 – 4 2009.

[16] W. E. Winkler. The state of record linkage and current research problems. Technical report, Statistical Research Division, U.S. Bureau of the Census, 1999.

[17] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques.* Morgan Kaufmann Publishers, 2005.