

How to build a QA system in your back-garden: application for Romanian

Constantin Orăsan
Computational Linguistics Group
University of Wolverhampton
C.Orasan@wlv.ac.uk

**Doina Tatar, Gabriela Șerban,
Dana Lupsa and Adrian Oneț**
Faculty of Mathematics and Computer Science
Babeș-Bolyai University
{dtatar, gabis, davram,
adrian}@cs.ubbcluj.ro

Abstract

Even though the question answering (QA) field appeared only in recent years, there are systems for English which obtain good results for open-domain questions. The situation is very different for other languages, mainly due to the lack of NLP resources which are normally used by QA systems. In this paper, we present a project which develops a QA system for Romanian. The challenges we face and decisions we have to make are discussed.

1 Introduction

Question answering (QA) emerged in the late 90s as a result of the Text Retrieval Conferences (TREC). These conferences are designed to evaluate the state-of-the-art in text retrieval and allow the participants to evaluate their systems in a consistent way by providing them a common test set. Starting with TREC-8, in 1999, these conferences contain a question answering track in which the participants try to find the answer to questions in a large collection of texts. As a result of the TREC, the QA field witnessed rapid development for English, but there are only few systems which work for languages other than English (Kim and Seo, 2002; Vetulani, 2002). Another factor which slows down the development of QA systems for other languages than English is the lack of the modules which are normally used in a QA system.

In this paper, we discuss the challenges we have to face during the development of a question

answering system for Romanian language. This paper is structured as follows: In Section 2, we present the structure of our QA system. The problems which need to be tackled when it is implemented for Romanian are presented in Section 3. A discussion of the project is presented in Section 4, the article finishing with conclusions.

2 The structure of our question answering system

A QA system normally contains three modules: *a question processor*, *a document processor* and *an answer extractor module* (Harabagiu and Moldovan, 2003). In addition, QA systems also rely on a generic or specially designed search engine. Our system follows this structure.

The question processor transforms a natural language question in an internal representation which can be a list of keywords or some kind of logical form. The list of keywords can contain only words from the question, or it can be expanded with words related to the ones in the question. Even if the question is transformed to a more advanced representation than a simple list of keywords, given that the QA systems rely on search engines, it is necessary to produce a list of keywords which are used to query the search engine. The more advanced form is used by the answer extraction module to locate the answer.

At present, our system extracts only keywords from the question. These keywords are expanded with semantically related words in the way presented in Section 3.3. Currently, we are considering using partial parsing trees for representing the structure of the question in a manner similar to (Buchholz and Daelemans,

2001). Unfortunately, to the best of our knowledge, there are no partial parsers for Romanian, so if such a method is to be tried, we will have to implement a partial parser first.

Another role of the question processor is to identify the type of the answer required by a question. Usually patterns are used to recognise this type. After analysing questions produced by experts, we compiled a list of patterns which trigger certain types of answers. Some of these patterns are presented in the Table 1.

A list of text snippets which could contain the answer to the asked question is obtained by querying a search engine with the keywords produced by the question processor. In our research, we use the `ht://Dig`¹, a tool which can be used to index and search medium size collections of documents and which behaves in a similar way with most of the search engines.

The document processor applies different NLP techniques to the extracted snippets. These techniques range from simple part of speech tagging to advanced ones such as coreference resolution and word sense disambiguation. One of the modules which are essential for any QA system is the named entity recogniser. Its role is to ensure that the extracted answer contains the type of entity required by the question.

In some cases the document processor reorders the snippets on the basis of the words contained in them. Pasca and Harabagiu (2001) show that such an approach has significant influence on the overall performance of the QA systems.

The answer extraction module locates the required answer in the list of text snippets extracted by the search engine, taking into consideration several factors. First, the answer has to contain an entity of the type specified in the question. Other factors which are considered when a text snippet is selected as containing the answer are the distribution of the keywords in the snippet and their frequency. In most cases, in addition to the keywords contained in the query, semantic variations and coreferential words are used to compute these statistics. The TF-IDF scores of the keywords are also employed to

determine the relevant answer.

3 Problems

In the previous section, we showed that the QA systems usually rely on a large number of NLP tools in order to achieve their goals. For less researched languages, such as Romanian, these tools are not available. In this section, we show how we addressed the problem of lack of resources.

3.1 The data

The open-domain question answering systems usually operate on the Web or on large collections of data which are meant to replace it. Unfortunately, the number of web pages in Romanian is quite negligible in comparison with the ones in English. Several search engines allow to retrieve only pages in a language which is specified, but their results are not always reliable. In light of this, we decided that in the initial stages of the project, we should locate the answers in a collection of documents available on a local machine. Our collection consists of newspaper articles published in two Romanian newspapers (*Evenimentul Zilei*² and *Adevarul*³). The articles were automatically downloaded and converted to plain text format. At present the collection of documents totalises over 12mil. words. In later stages of the project, we intend to try the system on the Web, even though this could raise additional problems.

3.2 The search engine

In order to make the future transition from our local collection to the Web, we needed to use a search engine which operates in a very similar manner with those which index the Web. As already mentioned, we use the freely available `ht://dig` tool. An advantage of using this tool, is that we can control the properties of the retrieved text snippets (e.g. length).

3.3 Using ontologies

One of the most used resources by QA system are ontologies, such as WordNet. The version

¹Available at: <http://www.htdig.org>

²<http://www.expres.ro>

³<http://adevarul.kappa.ro>

Pattern	Question	Type of answer
CINE X ?	Cine este presedintele Romaniei? <i>Who is the president of Romania?</i>	PERSON
UNDE X ?	Unde se afla Mariana Stanciu in 17 aprilie? <i>Where was Mariana Stanciu on the 17th April?</i>	LOCATION
CE BUILDING X?	Ce castel este faimos in Romania? <i>Which castle is famous in Romania?</i>	LOCATION
CAND X?	Cand a fost adoptata Constitutia? <i>When was the Constitution adopted?</i>	DATE

Table 1: Some questions which can be answered by our system

for Romanian WordNet is currently in the early stages of development, so we had to find a way to replace it. Given that we did not have the resources to build an ontology by hand, we decided to use unsupervised methods which cluster words together according to the context in which they appear. The clusters indicate that the words are semantically related.

Two clustering algorithms have been implemented and tested. The first one is a non-hierarchical clustering which starts with several random clusters which are, then, refined. The second clustering algorithm is a bottom-up hierarchical clustering algorithm. Evaluation of the results showed that the hierarchical algorithm is more accurate for the task (Tatar and Şerban, 2003). Figure 1 shows few of the clusters we obtained.

- Cluster 1** *timp, partid, persoana, sat*
- Cluster 2** *oras, localitate*
- Cluster 3** *durata, perioada*
- Cluster 4** *oameni, organizatie, asociatie*

Figure 1: Few of the obtained clusters

3.4 Named entity recognition

The task of named entity recognisers is to identify phrases which refer to people, places, organisations, etc. As with many other fields, most of the available tools are for English, but the CoNLL02 shared task (Tjong Kim Sang, 2002) has shown that it is possible to use machine learning approaches to design

named entity recognisers for languages other than English. However, these approaches need annotated corpora to learn how to identify the named entity.

Named entities in more than 100 articles were marked using our multi-purpose annotation tool.⁴ These files will be used to train several machine learning algorithms which identify the named entities, and the best performing one will be included in the QA system.

3.5 Other tools we used

In addition to the tools and resources previously enumerated, we had to develop some basic tools which one expects to find in any language. All our attempts to locate a tokenizer and a stemmer failed. A large number of tools and resources were developed by the Multext Project⁵, but unfortunately they are no longer available. Even though the development of these tools is not difficult, we want to emphasise that when beginning such a project for Romanian, it is necessary to start with very simple resources and tools.

We also used the TnT part-of-speech tagger (Brants, 2000) with the language models for Romanian described in (Tufis, 2000) to tag the questions and the text snippets.

4 Discussion

In the previous sections, we showed the structure of our question answering system and how we

⁴Available at: <http://elg.wlv.ac.uk/projects/PALinkA/>

⁵<http://www.lpl.univ-aix.fr/projects/multext>

replaced missing components with knowledge poor methods. For each of them several alternative algorithms will be implemented and the best performing combination will be included in the final system. As a result of this project several tools for Romanian will be developed.

As can be noticed, the structure of our QA system is the same as any English QA system. One question which we will try to answer in this project is how much the QA systems are language dependent. We will investigate what kind of components are required for the Romanian language, in addition to those included in English systems.

Given the nature of the Romanian language, we expect that some of the components will perform better than their equivalents for English and that they will provide more information. For example, if a coreference resolver will be included in the system, we expect to be able to obtain high accuracy thanks to the stricter agreement in Romanian.

When all the components will be fully implemented, the system will be evaluated using the TREC methodology. In order to evaluate the system we asked experts to read the newspaper articles and propose factual questions which can be answered using short texts from the articles. In addition to the human directed evaluation, we are planning to have also automatic evaluation. For this reason we asked our experts not only to propose questions, but also to indicate which is the expected answered.

5 Concluding remarks

In this paper, we presented an ongoing project which develops a question answering system for Romanian. Even though the structure of our system does not bring new features, its novelty consists in the fact that this is the first QA system for Romanian. The existing gaps in the list of available resources for Romanian were filled in by employing knowledge-poor methods which require little or no training data.

The explanation of the title “How to build a QA system in your back-garden” is that should this project be successful, it will provide not only a QA system for Romanian but it will also prove that it

is possible to develop QA systems for less studied languages without the need of many resources.

References

- Thorsten Brants. 2000. TnT - a statistical part-of-speech tagger. In *Proceedings of the Sixth Conference on Applied Natural Language Processing (ANLP-2000)*, Seattle, WA.
- Sabine Buchholz and Walter Daelemans. 2001. SHAPAQA: shallow parsing for question answering on the World Wide Web. In *Proceedings of RANLP'2001*, pages 47 – 51, Tzigrav Chark, Bulgaria, 5 – 7 September.
- Sanda Harabagiu and Dan Moldovan. 2003. Question answering. In Ruslan Mitkov, editor, *Oxford Handbook of Computational Linguistics*, chapter 31, pages 560 – 582. Oxford University Press.
- Harksoo Kim and Jungyun Seo. 2002. A reliable indexing method for a practical QA system. In *Proceedings of the Workshop on Multilingual Summarisation and Question Answering*, pages 17 – 24, Taipei, Taiwan, August 31st – September 1st.
- Marius Pasca and Sanda Harabagiu. 2001. High performance question/answering. In *Proceedings of the 24th Annual International ACL SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2001)*, pages 366 – 374, Toulouse, France.
- Doina Tatar and Gabriela Șerban. 2003. Words clustering in question answering systems. *Studia Universitatis Babeș-Bolyai, Series Computer-Science*, XLVIII(2).
- Erik F. Tjong Kim Sang. 2002. Introduction to the CoNLL-2002 Shared Task: Language-independent named entity recognition. In *Proceedings of the Sixth Conference on Natural Language Learning (CONLL-2002)*, Taipei, Taiwan, August 31 – September 1.
- Dan Tufis. 2000. Using a large set of EAGLES-compliant morpho-syntactic descriptors as a tagset for probabilistic tagging. In *Proceedings of the Second International Conference on Language Resources and Evaluation*, pages 1105 – 1112, Athens, Greece, May.
- Zygmunt Vetulani. 2002. Question answering system for Polish (POLINT) and its language resources. In *Proceedings of the Question Answering - Strategy and Resources Workshop*, pages 51 – 55, Las Palmas de Grand Canaria, 28th May.