University of Wolverhampton at CLEF 2008

Iustin Dornescu, Georgiana Puşcaşu and Constantin Orăsan
University of Wolverhampton at CLEF 2007
{I.Dornescu2, georgie, C.Orasan}@wlv.ac.uk

Abstract

This article presents the participation of University of Wolverhampton in the Romanian to English Question Answering task at CLEF-2008. This year we employed a modular framework which allows different modules to be easily plugged in and customised. The main components of our system deal with the three standard stages used in question answering: question processing, paragraph retrieval and answer extraction, and the system's cross-linguality is ensured by a term translator. The question processor analyses Romanian questions and produces a detailed representation of each question including the terms it contains. English translations are then generated for all question terms by exploiting information included in the Romanian and English WordNets, as well as aligned Wikipedia pages. They form the query that Lucene uses to extract English paragraphs which constitute the input for an answer extractor largely based on the one distributed with the OpenEphyra framework. The results indicate a small improvement in comparison with last year's performance.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.1 Content Analysis and Indexing; H.3.3 Information Search and Retrieval; H.3.4 Systems and Software; H.3.7 Digital Libraries; H.2.3 [**Database Managment**]: Languages—*Query Languages*

General Terms

Measurement, Performance, Experimentation

Keywords

Question answering, Questions beyond factoids

1 Introduction

This paper describes the second participation of the University of Wolverhampton in the Romanian-English cross-language QA@CLEF competition. The objective for this year was to develop a question answering (QA) framework in which different modules are plugged in dynamically so that different processing components can be included as needed. This framework also allows individual module tuning and performance assessment, and can easily be developed further. Our architecture relies on OpenEphyra, a modular and extensible framework for question answering [5], which was the basis for some TREC submissions [8]. Whilst our objective of having a modular system was achieved, due to time restrictions, some of the system components are in the early stages of development, which has led to a relatively small performance increase compared to last year's results [7].

Our system adheres to the established pipeline architecture for QA that consists of a question processor, a passage retrieval module and an answer extractor [2]. Similar to the approach taken in previous years, our system relies on an intermediate stage that translates questions processed in the source language and feeds its output into the answer extractor that retrieves passages and locates exact answers in the target language. In this way it is possible to have questions asked in Romanian and answers extracted from an English collection. The rest of the paper describes the system components and the evaluation results. Sections 2 to 5 describe in turn each of the four stages involved in the cross-lingual QA process: question processing (section 2), term translation (section 3), passage retrieval (section 4) and answer extraction (section 5). Section 6 captures the results achieved and an error analysis. Conclusions and directions of future work tackling the problems encountered during this year's participation are presented in section 7.

2 Question Processing

At this first stage of the system, each Romanian question is analysed with the aim of identifying relevant information necessary in the subsequent stages of the system's answering process. The question processing module produces a custom representation of the question that includes: the question focus, the expected answer type (EAT), the question type, the relevant terms identified in the question and the question topic. The way each of these is determined is explained below.

The question processor employed this year is an improved version of the one embedded in last year's system. As before, this module involves several preprocessing stages such as part-of-speech, noun and verb phrase identification, numerical and temporal expression annotation, all improved to address problems identified during last year's participation in QA@CLEF. This year's system uses a revamped phrase and named entity recogniser based on dictionaries and entity mappings extracted from the aligned Romanian and English Wikipedias as described in Section 3.

The linguistic information obtained during preprocessing is used to identify the question focus, which is important in the search for an answer, as it normally reveals what the question is asking for or what the question is about.

The semantic type of the expected answer is then determined by matching question focus and the first question verb phrase against syntactic constraints and semantic hierarchies extracted from WordNet (e.g. the noun $arhitect_{ro}$/$architect_{en}$, as hyponym of the synset *person, individual, someone, somebody, mortal, soul* in WordNet, is automatically mapped to the EAT category PERSON). As we did last year, predefined mappings between categories of expected answers and WordNet synset hierarchies are employed. A more detailed EAT categorisation is covered by this year's question processing module and includes the following classes: PERSON, LOCATION, ORGANIZATION, NAME, TYPE, NATIONALITY, LANGUAGE, OCCUPATION, DEFINITION, NUMERIC (with the sub-classes QUANTITY, MEASURE, ECONOMIC, PERCENTAGE) and TEMPORAL (with the sub-classes CENTURY, YEAR, MONTH, WEEK, DATE, TIME, DURATION). The number of categories employed by our system was increased after noticing that a number of questions from last year's test set could not be answered due to the fact that we had too general categories.

As in previous editions, this year's questions are concerned with facts or events (FACTOID questions), definitions of people, organisations or things (DEFINITION questions), or lists of people, objects or dates (LIST questions). These question types are recognised using the same approach as last year [7].

The question processor also produces a list of keywords in decreasing order of their relevance. The list contains noun and verb phrases, named entities, temporal and numeric expressions that appear in the question to be answered, and is used as the input of the term translation module. The topic corresponding to a cluster of questions, whenever it can be identified following the procedure described below, is also added to the keyword list of each question in the cluster.

The fact that questions are grouped in clusters related to the same topic is exploited by the question processor to improve the results of the system. This topic is usually present in the first question or represents the answer to the first question. Due to the fact that the current architecture does not allow us to feed answers back into the system, we consider as topic the first named entity of type PERSON that appears in the first question of each cluster. If no such entity can be found, the first entity of the question is selected as the topic. If the question contains no named entities, the topic is considered to be the question focus. This approach was designed after empirical analysis of last year's questions. The topic is used to boost the retrieval scores of the Wikipedia articles describing it.

3 Term Translation

The cross-linguality of the system is ensured by a term translation module that takes as input the keywords identified during question processing and generates a ranked list of translation equivalents. Firstly, a list of translation equivalents mined from the mappings between the Romanian and English Wikipedias is used in order to obtain high quality translations of entities. This method provides a high precision, but low recall term translation.

The Inter-Lingual Index (ILI) between the Romanian [9] and the English [1] WordNets is used to obtain all translation equivalents for the remaining keywords. If a word does not appear in the Romanian WordNet, alternative dictionaries are consulted. The drawback to this method is that it yields too many translation candidates. For this reason, a ranking method that relies on co-occurrence of words in Wikipedia is used to filter out infrequent candidates. This method proved particularly useful for translating noun phrases, where all the combinations of word by word translations were generated and sought in Wikipedia. The infrequent ones were removed, as they indicated incorrect translations.

For example, given the Romanian term *plan general*, the translation equivalents generated using the Romanian and English Wordnets are the following:

general plan, general plane, general mind, general idea, general program, general sheet, general design, general cadre, general canvas, general programme, general inclined plane, general architectural plan

The ranking method then identifies the words that most frequently appear together in Wikipedia and eliminates those with infrequent use, and the resulted translations are:

general design, general plan, general program, general idea, general programme

4 Passage Retrieval

The purpose of this module is to provide the answer extractor with sentences from the document collection that are relevant to the question. This is achieved by using the identified topic and the translated terms. A two stage retrieval approach is employed: first, the most relevant documents are selected, and then all their sentences are extracted and re-ranked according to their relevance to the query. In order to do this the corpus was preprocessed and indexed using the Lucene retrieval engine [4].

4.1 Corpus preprocessing and indexing

As last year, the answers had to be extracted from a heterogeneous collection consisting of two news corpora: Los Angeles Times from 1994 and Glasgow Herald from 1995, as well as English Wikipedia pages from November 2006. Wikipedia can be processed either as a static HTML dump, or in its native wikisource format. The former has a lot of content that is not part of the article itself and could have a negative impact on the answer extraction process, whereas the later is easier to process as it contains less information irrelevant to our purposes. After analysing the advantages and disadvantages of each format, we decided to convert the wikisource dump to plain text, preserving the information included in infoboxes, lists and tables, together with their markers. The inter-language links between the Romanian and English Wikipedias were also preserved in order to create the Romanian to English bilingual entity dictionary that was employed by the term translation module (see Section 3).

The text of the articles was indexed using a standard approach. No stopword filtering was employed, since stopwords are important for question answering. The index was enriched with a stemmed version of the text obtained using Porter Stemmer [6]. This was done because retrieval using stemming offers higher recall, whereas retrieval using full words offers greater precision. Our retrieval engine used a combination of the two available in Lucene.

4.2 First stage - document retrieval

The first retrieval stage uses the translated query to extract documents which might contain the answer to a question. As we previously mentioned, our system relies heavily on information extracted from Wikipedia. To this end, we first try to identify documents related to the topic of the question by retrieving documents which contain the question topic in their title. In cases where the topic is not reliably identified or no documents that contain the topic in their title can be found, documents which contain the query terms are retrieved.

In order to create complex queries that would rank higher the documents describing the target entities, we use a combination of phrase, fuzzy and proximity queries, as well as term boosting techniques offered by Lucene. A maximum of 5 documents from the query result set are considered.

4.3 Second stage - sentence selection

The second stage of paragraph retrieval tries to identify sentences which are relevant to the question. In order to do this, the documents retrieved at the previous stage are segmented into sentences using LingPipe [3] and used to create an in-memory index. This index is queried using the keywords produced by the term translation module in order to extract up to 50 sentences which are passed to the answer extractor.

5 Answer extractor

The answer extractor employed by our system relies heavily on the answer extraction modules provided by OpenEphyra. These modules implement several answer extraction strategies depending on the type of question, source of answer, and above all, expected answer type. In order to employ these components, our EAT hierarchy had to be mapped to the one used by OpenEphyra. In this way, most of the answers retrieved by our system are named entities of the type given by the expected answer type. This approach yielded good results for the categories identified by usual named entity recognisers (i.e. LOCATION, ORGANIZATION, PERSON, NUMBER, DATE), but proved unsuitable for questions requiring GENERIC answers such as *Numiti doua instrumente la care canta Emerson./Name two instruments played by Emerson.* Due to time restrictions, no attempt was made to change the ranking algorithm implemented by OpenEphyra. For this reason, in a large number of cases, the correct answer was among the first three answers retrieved by the system, but not the top one.

Definition questions are answered using a different answer extraction module from the one provided by OpenEphyra. Our module relies on a cascade of high precision filters designed after analysis of the data sources and experience gained in previous CLEF participations.

Wikipedia is a great source for definitions. For this reason, whenever we try to find a definition for a given term, Wikipedia is the first place to look for one. We start by locating Wikipedia pages which contain the term to be defined in their title. In some cases this approach fails because the title of the page has a different surface form even though it refers to the same concept (e.g. the page about *CORGI* has the title *Council for Registered Gas Installers*). In these cases, we check whether the term to be defined is used as the name of a file (i.e. considering the previous example we check whether there is a Wikipedia file called *CORGI*). Once candidate pages are located, a set of patterns is applied in order to extract the definition of the term. The patterns are designed in such a way that they can cater for situations where alternative forms are used to refer to the same concept (e.g. *Steve Redgrave* is referred to in an Wikipedia article as *Sir Stephen Geoffrey Redgrave*). The definition of a term is considered to be the whole sentence to which a pattern can be applied.

If no definition can be located in a Wikipedia page using the method described above, we then search the whole collection for sentences that contain not only the term to be defined, but also other terms that might appear in a DEFINITION question (sometimes the question provides disambiguation clues for the term to be defined, e.g. *Ce este "bungo" in japoneza?/What is "bungo" in Japanese?*). A different set of manually created patterns is applied to these sentences in order to extract the definition of the term. In contrast to the definitions extracted by the previous method, at this stage we extract only the noun phrase which is considered to define a term (e.g. *Richard D. Farman* is defined as *CEO of Southern California Gas Co.*).

6 Evaluation results

The evaluation results reveal an improvement in comparison with last year's accuracy. They are presented in Table 1.

Table 1: Official results

Question Type	Right	Inexact	Unsupported	Wrong	Accuracy
Definition	20	1	0	9	66.67%
Factoid	18	1	5	136	11.25%
Lists	0	0	0	10	0.00%
Overall	38	2	5	154	18.00%

As expected, the best results are obtained for definition questions which can easily be answered thanks to the structure of Wikipedia pages. The main source of errors in the case of definition questions is the incorrect translation of the term to be defined and errors introduced by preprocessing tools. For example, one definition was marked as inexact due to the fact that the NP extractor wrongly identified a bigger chunk of text which included the NP that constituted the answer. Translation errors also contributed to wrong answers or no answers being extracted for factoid questions. However, the main source of errors was the fact that no mapping between the GENERIC expected answer type and the named entity classes covered by OpenEphyra was used. As a result, all the questions expecting GENERIC answers were answered with NIL.

A large number of questions with NUMERIC EAT were wrongly answered due to errors in the named entity recogniser employed by OpenEphyra. Numeric parts of date expressions were wrongly labelled as numbers and returned as answers (e.g. the number *12,1853* is wrongly extracted as answer from the date *March 12, 1853*). In many cases the correct answer was extracted as well, but received a lower confidence.

Currently our system does not have a way to deal with list questions and for this reason it returned NIL for all these questions.

The modular structure of our system also enables us to assess the performance of each individual component. The question analysis module identifies the question type with an accuracy of 98% and the expected answer type with 94%.

Empirical observation of the translation output indicates that there are still issues to be addressed in the future such as the ranking of the translation equivalents and translation of named entities especially when the questions contain words from several languages (e.g. *La ce data a scris Mathieu Orfila al sau "Tráite des poisons"?/When did Mathieu Orfila write his "Tráite des poisons"?*).

7 Conclusions

In this article we presented our participation in the Romanian to English Question Answering task at CLEF-2008. We employed a modular system consisting of the three standard stages of a QA system: question processing, paragraph retrieval and answer extraction. The question processor analyses Romanian questions and produces a detailed representation of each question including the terms it contains. These terms are then translated to English using several techniques based on the Romanian and English WordNets and aligned Wikipedia pages. Lucene is used to extract English paragraphs which constitute the input for the answer extractor employed. Due to time restrictions, some of the modules are in early stages of development or have been adapted from other projects. For example, the answer extractor is largely based on the one distributed with the OpenEphyra framework.

The existing components of OpenEphyra were designed to process English questions and retrieve answers from English document collections. For the future, we plan to use existing translation services such as Google Translate to obtain full translations of the Romanian questions as an additional source of information for our system. These translations can also be used by a monolingual English QA system such as OpenEphyra. Comparison between the two approaches can give us further insights into the best approach for cross-lingual question answering, and how and whether they can be combined.

8 Acknowledgements

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

References

[1] Christiane Fellbaum, editor. *WordNet: An Eletronic Lexical Database.* The MIT Press, 1998.

[2] Sanda Harabagiu and Dan Moldovan. Question Answering. In Ruslan Mitkov, editor, *Oxford Handbook of Computational Linguistics*, chapter 31, pages 560 – 582. Oxford University Press, 2003.

[3] LingPipe. http://alias-i.com/lingpipe/.

[4] LUCENE. http://lucene.apache.org/java/docs/.

[5] OpenEphyra. http://sourceforge.net/projects/openephyra/.

[6] Martin F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130 – 137, 1980.

[7] Georgiana Puşcaşu and Constantin Orăsan. University of Wolverhampton at CLEF 2007. In *Working Notes for the Cross Language Evaluation Forum (CLEF) 2007 Workshop*, Budapest, Hungary, 2007.

[8] Nico Schlaefer, Jeongwoo Ko, Justin Betteridge, Guido Sautter, Manas Pathak, and Eric Nyberg. Semantic Extensions of the Ephyra QA System for TREC 2007. In *Proceedings of the Sixteenth Text REtrieval Conference (TREC)*, 2007.

[9] Dan Tufis, Dan Cristea, and Sofia Stamou. BalkaNet: Aims, Methods, Results and Perspectives. A General Overview. In D. Tufis, editor, *Romanian Journal on Information Science and Technology. Special Issue on BalkaNet.* Romanian Academy, 2004.