Developing a Question Answering System for the Romanian-English Track at CLEF 2006

Georgiana Puşcaşu[†§♮], Adrian Iftene[†], Ionuţ Pistol[†], Diana Trandabăţ[†], Dan Tufiş[*], Alin Ceauşu[*], Dan Ştefănescu[*], Radu Ion[*], Constantin Orăsan[§], Iustin Dornescu[†], Alex Moruz[†], Dan Cristea[†]
[†]Faculty of Computer Science
"Alexandru Ioan Cuza" University, Romania
{adiftene, ipistol, dtrandabat, idornescu, amoruz, dcristea}@infoiasi.ro
[*] Centre for Artificial Intelligence
Romanian Academy, Romania
{tufis, alceausu, danstef, radu}@racai.ro
[§] Research Group in Computational Linguistics
University of Wolverhampton, UK
{georgie,C.Orasan}@wlv.ac.uk
[♮] Department of Software and Computing Systems
University of Alicante, Spain
{georgie}@dlsi.ua.es

Abstract

This paper describes the development of a question answering system for the Romanian-English cross-lingual track organized within the Cross Lingual Evaluation Forum (CLEF) 2006 campaign. The development stages of our cross-lingual Question Answering (QA) system are described incrementally throughout the paper, at the same time pinpointing the problems that occurred and the way they were addressed. Our system adheres to the classical architecture for QA systems, debuting with question processing followed, after term translation, by information retrieval and answer extraction. Besides the common QA difficulties, the track posed some specific problems, such as the lack of a reliable translation engine from Romanian to English, and the need to evaluate each module individually for a better insight into the system's failures.

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;; I.2 [**Artificial Intelligence**]: I.2.7 Natural Language Processing

General Terms

Measurement, Performance, Experimentation

Keywords

Question Answering, Cross-lingual Question Answering

1 Introduction

Question Answering can be defined as the task which takes a question in natural language and produces one or more ranked answers from a collection of documents. The QA research area has really emerged as a result of the introduction of a monolingual English QA track within the Text Retrieval and Evaluation Conference (TREC). Multilingual, cross-lingual, as well as monolingual QA for languages other than English are addressed at a scientific level by the CLEF evaluation campaigns [7].

This year our team[1] of students, PhD students and researchers, have taken part for the first time in the QA@CLEF competition. As in every other cross-lingual task, the system input consisted of 200 questions in Romanian, and the output was expected to be the exact English answer. As this was the first time the Romanian-English track was introduced at CLEF, most of the effort was directed towards the development of a fully functional cross-lingual QA system having as source language Romanian, and less on fine-tuning the system to maximize the results.

Following the generic architecture for QA systems [2], our system contains a question analyzer, an information retrieval engine and an answer extraction module, as well as a cross-lingual QA specific module which translates the relevant question terms from Romanian into English.

This paper describes the steps taken in the development of our cross-lingual QA system for the Romanian-English track at CLEF 2006, as well as its evaluation. The remainder of the paper is structured as follows: Section 2 provides a description of the system and its embedded modules, while Section 3 presents the details of the submitted runs. Section 4 comprises an analysis of the results obtained, and finally, in Section 5, conclusions are drawn and future directions of system development are considered.

2 System Description

2.1 System Overview

Question Answering systems normally adhere to the pipeline architecture that consists of three main stages: question analysis, paragraph retrieval and answer extraction [2]. Specific systems can be seen as instantiations of the general architecture, with particular choices being made concerning representation and processing for each component of the overall model. The first stage is question analysis. The input to this stage is a natural language question and the output is one or more representations of the question to be used in subsequent stages. At this stage most systems identify the semantic type of the entity sought by the question, determine additional constraints on the answer entity, and extract the keywords to be employed at the passage retrieval stage. The next stage, passage retrieval, is typically achieved by employing a conventional IR search engine to select a set of relevant candidate passages from the text collection. At the last stage, answer extraction and ranking, the representation of the question and the representation of the candidate answer-bearing passages are compared and a set of candidate answers is produced, ranked according to how likely they constitute the correct answer. Apart from the typical modules included in a QA system, our system also includes a module which translates terms from Romanian into English, in order to perform the cross-lingual transfer. The system architecture and functionality are illustrated in Figure 1.

2.2 NLP Pre-processing

The questions are first morpho-syntactically pre-processed using the Romanian POS tagger developed at RACAI [8]. Afterwards, a pattern-based Named Entity Recognizer is employed to

[1] The team that participated in the development of the system included students, PhD students and researchers from the Faculty of Computer Science at the "Alexandru Ioan Cuza" University, Iasi, Romania (furtherly addressed as UAIC), from the Centre for Artificial Intelligence of the Romanian Academy - (RACAI), from the University of Alicante, Spain, and the University of Wolverhampton, United Kingdom.

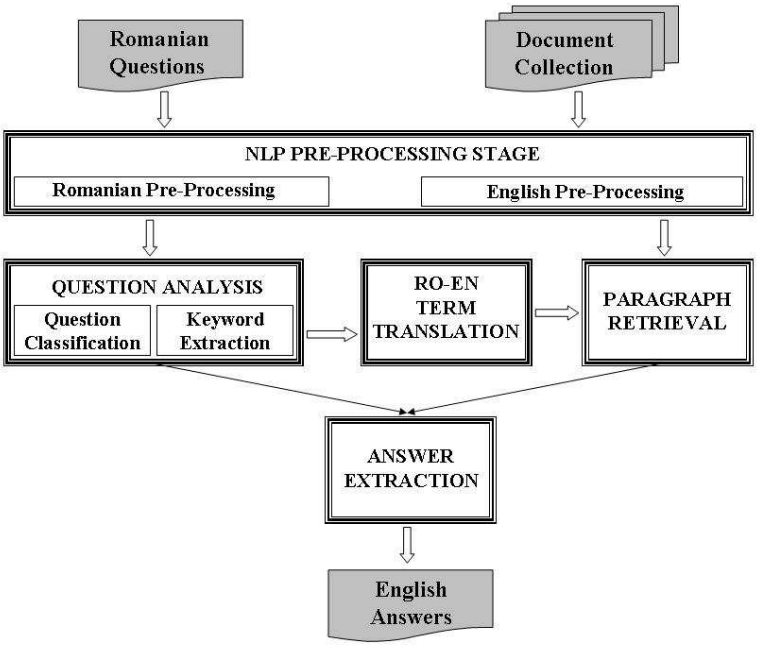

Figure 1: System Architecture and Functionality

identify Named Entities classified in one of the following classes: *Person*, *Location*, *Measure* and *Miscellaneous*.

Similar pre-processing operations are carried out on the English document collection, after segmenting it at sentence level. We used the same set of tools with a different language model.

2.3 Question Analysis

This stage is mainly concerned with the identification of the semantic type of the entity sought by the question (expected answer type). In addition it also identifies the question focus, the question type and the set of keywords relevant for the question. To achieve these goals, our question analyzer performed the following steps:

a) ***NP-chunking, Named Entity extraction, Temporal Expression identification***
The input at this stage consists of the pre-processed set of questions. On the basis of the morpho-syntactic annotation provided by our Romanian POS-tagger, we have implemented a rule-based shallow noun phrase identifier. The Named Entity recognizer employed at the pre-processing stage provides us with the set of question Named Entities. Temporal expressions (TEs) are also identified using the adaptation to Romanian of the TE identifier and normaliser developed by [6].

b) ***Question focus identification***
The question focus is the word or word sequence that defines or disambiguates the question, in the sense that it pinpoints what the question is searching for or what it is about. We considered the question focus to be either the noun determined by the question stem (as in *What* ***country***) or the head noun of the first question NP if this NP comes before the main verb of the question or if it follows the verb "*to be*".

c) ***Distinguishing the expected answer type***
At this stage we identify the category of the entity expected as an answer to the analyzed question. Our system's answer type taxonomy distinguishes the following classes: PERSON,

LOCATION, ORGANIZATION, TEMPORAL, NUMERIC, DEFINITION and GENERIC. The assignment of a class to an analyzed question is performed using the question stem and the type of the question focus. The question focus type is detected using the WordNet [1] sub-hierarchies specific to the categories PERSON / LOCATION / ORGANIZATION. We manually attach to each category a set of ILI numbers representing the root nodes of the WordNet sub-trees containing category-specific nouns. Using the ILI numbers, we extract from the Romanian WordNet [9] lists of nouns for each of the three categories. In the case of ambiguous question stems (e.g. *What*), we search in the resulted lists for the head of the question focus, and identify as expected answer type the category of the corresponding list (for example, in the case of the question *In which city was Vladislav Listyev murdered?*, the question focus is *city*, noun which appears in the LOCATION list, therefore the associated expected answer type is LOCATION).

d) ***Inferring the question type***
This year, the QA@CLEF competition has distinguished among four types of questions: *factoid*, *definition*, *list* and *temporally restricted* questions. As temporal restrictions can constrain any type of question, we proceed by first detecting whether the question has the type *factoid*, *definition* or *list* and then test the existence of temporal restrictions. The question type is identified using two simple rules: if the expected answer type is DEFINITION, then obviously the question type is *definition*; if the question focus is a plural noun, then the question type is *list*, otherwise *factoid*. The temporal restrictions are identified using several patterns and the information provided by the TE identifier.

e) ***Keyword set generation***
The set of keywords is automatically generated by listing the important question terms in decreasing order of their relevance. Therefore, the set of keywords comprises: the question focus, the identified NEs and TEs, the remaining noun phrases, and all the non-auxiliary verbs present in the question. This set is then passed on to the Term Translation module, in order to obtain English keywords for paragraph retrieval.

2.4 Term Translation

In order to achieve term translation we have employed WordNet, a resource available both for English and Romanian. The set of keywords extracted at the question analysis stage served as input for the term translation stage, therefore we were supposed to translate both noun phrases and verbs. The noun phrases are translated in a different manner than the verbs. The NP words are translated individually, by first identifying the Romanian synsets containing the word, by reaching through the ILI numbers the corresponding English synsets, and by forming a set of all possible translations. From this set, using empirical rules based on word frequency, we choose a maximum of three candidates. If the word to be translated does not appear in the Romanian WordNet, as it was quite frequently the case, we search for it in other available dictionaries and preserve the first three translations. If still no translations are found, we consider the word itself as translation. After each individual word is translated, we employ rules that translate the Romanian syntax into English syntax. In this manner we obtain for each NP several translation equivalents.

In the case of verbs, we extract for each verb the translation equivalents from WordNet as we did for nouns. Attempts to apply the frequency-based methodology used in the case of nouns failed due to the fact that some very general verbs were preferred instead of the correct translation equivalent. To this end, we decided to select the best translation equivalent by considering both the frequency of the verb and of the nouns which appear in the subject and object positions. To achieve this, the verb-noun co-occurrence data described in [5] has been used to determine which of the verb translation equivalents co-occur more often with the translated nouns, and was selected as the translation. Despite the simplicity of the method, the accuracy of the translation improved dramatically over the row frequency.

2.5 Index Creation and Paragraph Retrieval

As described in section 2.2, the English corpus was initially preprocessed using tokenization, lemmatization, POS-tagging and NE recognition tools. In our runs, we employed an index/search engine based on the Lucene [4] search engine.

The document collection was indexed both at document, as well as paragraph level, using the lemmas of the content words and the NE classes (MEASURE, PERSON, LOCATION, etc). For a query like:

```
(HOW MANY, MEASURE)
(record book "record book") AND (sport athlete sportsman) AND (world worldwide global)
AND (man husband male) AND (score hit carry) AND (1995)
```

the search engine searches for a segment (document/paragraph) that contains a measure and words belonging to the set of translation equivalents. The question type is used to approximate the maximum number of hits to be returned. When no paragraphs are returned for a given query, we employ two strategies: we either increase the granularity of the segments from paragraphs to documents, or we reformulate the query using for individual words spelling variations and variable word distance for phrases.

2.6 Answer Extraction

Two answer extraction modules have been developed, one by UAIC and another one by RACAI. Both modules require as input the expected answer type, the question focus, the set of keywords and the retrieved snippets together with their POS, lemma and NE information, and the relevance score returned by Lucene. The extraction process depends on whether the expected answer type is a Named Entity or not. When the answer type is a Named Entity, the answer extraction module identifies within each retrieved sentence Named Entities with the desired answer type. Therefore, in this case, the answer extraction process is greatly dependent on the results of the NE recognition module. When the answer type is not a Named Entity, the extraction process mainly relies on the recognition of the question focus, as in this case the syntactic answer patterns based on the focus are crucial. Both modules use the same pattern-based extractor to address DEFINITION questions.

a) ***Answering questions asking about Named Entities***
When the question asks about a Named Entity such as MEASURE, PERSON, LOCATION, ORGANIZATION, DATE, the UAIC answer extractor looks for all the expressions tagged with the desired answer type. If several such expressions exist, we choose the closest to the focus in terms of word distance, if the focus is present in the sentence, otherwise the first one occurring in the analyzed sentence. When there is no named entity of the desired type, we generalize the search using synonyms of the focus extracted from WordNet.

The RACAI answer extractor computes scores for each snippet based on whether a snippet contains the focus, on the percentage of keywords or keyword synonyms present in the snippet, and on the snippet/document relevance scores provided by the search engine. The entities having the desired answer type are identified and added to the set of candidate answers. For each candidate answer, another score is computed on the basis of the snippet score, the distance to the question focus and to other keywords. The candidate answers having scores above a certain threshold are presented as final answers.

b) ***Answering questions looking for GENERIC answers***
When the expected answer type is not a Named Entity (the question has the GENERIC answer type), the UAIC answer extractor locates the answer within the candidate sentence using syntactic patterns. The syntactic patterns for identifying the answer include the focus

noun phrase and the answer noun phrase, which can be connected by other elements such as comma, quotation marks, prepositions or even verbs. These syntactic patterns always include the focus of the question. Therefore, the focus has to be determined by the question analysis module in order to enable the system to identify answers consisting of a noun or verb phrase.

The RACAI answer extractor employs also patterns to select answer candidates. These candidates are then ranked using a score similar to the one employed for answering questions asking about NEs.

c) ***Answering DEFINITION questions***
In the case of DEFINITION questions, the candidate paragraphs are matched against a set of patterns. Each possible definition is extracted and added to a set of candidate answers, together with a score revealing the quality of the pattern it matched. The set of noun phrases present in the paragraphs are also investigated to detect those NPs containing the term to be defined surrounded by other words (this operation is motivated by cases like *the Atlantis space shuttle*, where the correct definition for *Atlantis* is *space shuttle*). The selected NPs are added to the set of candidate answers with a low score. Then, the set of candidate answers is ordered according to the score attached to each answer and to the number of other candidate answers it subsumes. The highest ranked candidate answers are presented as final answers.

3 Description of Submitted Runs

We have submitted three different runs, with the following detailed description:

- **UAIC**
  This run was obtained by parsing and analyzing the questions, translating the keywords, retrieving relevant passages and finding the final answers using the UAIC answer extractor.
- **RACAI**
  This run is also obtained by parsing and analyzing the questions, keyword translations, passage retrieval, but the answers were localized using the RACAI answer extractor.
- **DIOGENE**
  Our third run, which unfortunately was not evaluated, was obtained by converting the results of our Question Analysis and Term Translation modules to the format required by the DIOGENE QA system [3], and then providing them as input to the DIOGENE IR and Answer Extraction modules.

Due to the high number of submitted runs having English as target language only the UAIC and RACAI runs were evaluated. In the remainder of this paper, the RACAI run will be referred to as System 1, and the name System 2 will be used for the UAIC run.

4 Results and Analysis

4.1 General Evaluation Results

The official evaluation results of Systems 1 and 2 are presented in Figure 2. Each submitted answer was evaluated as UNKNOWN, CORRECT, UNSUPPORTED, INCORRECT and INEXACT.

The high number of answers evaluated as UNKNOWN is due to the fact that we provided ten answers for almost all 200 questions, as ten was the maximum number of answers allowed. However, the final evaluation evaluated only the first answer for the majority of the questions (only in the case of *list* type questions the first three answers were evaluated). As a result of this, the answers ranked 2 to 10 were labelled as UNKNOWN indicating that no attempt was made to check their correctness. From our internal evaluation, we have noticed that the correct answers were found in the first ten answers generated by our systems for 35-40% of the questions. This

Evaluation results for System 1		
Z	Unknown	400
R	Correct	35
U	Unsupported	13
W	Incorrect	184
X	Inexact	7
	Total	639

Evaluation results for System 2		
Z	Unknown	543
R	Correct	22
U	Unsupported	4
W	Incorrect	191
X	Inexact	1
	Total	761

Figure 2: Evaluation results for the two evaluated runs

is a promising result, as it shows that the answer extractors work, but a better ranking of the produced answers is required.

Most of the UNSUPPORTED and INEXACT answers are caused by the lack of a perfect match between the answer string and the supporting snippet. The majority of these errors can be solved with the use of a knowledge base, that will allow, for example, the identification of *Atlantis* as a *space shuttle* (question 101). Also, sometimes our systems provide a more generic, or more specific answer than required, as it is the case for the gold answer *Shoemaker* contrasted to our answer *Carolyn Shoemaker*, marked as UNSUPPORTED. These errors could be corrected by improving the answer extractor with more specific rules as to the format of the required answer. Another type of error causing UNSUPPORTED and INEXACT answers are the *list* type questions. We provided one answer per line, even if the question was of the type *list*. Most of the correct answers were found in the 10 answers provided, but not on the same line. These errors could be solved with a better classification of the returned answers and by grouping them according to the characteristics of the required list. There are also cases when the answer is marked as UNSUPPORTED, but denotes the same entity as the gold answer. The answer appears in the indicated snippet, but the character string is not identical with the gold answer (for example System 2 identifies the answer *United States*, but is marked as UNSUPPORTED because the related snippet for this answer contains *United_States*). These snippet changes are introduced at the pre-processing stage.

4.2 Detailed Comparison between the two Evaluated Runs

The two runs/systems chosen for the official evaluation are the two completely designed and implemented by the two Romanian groups involved, UAIC and RACAI. These two runs were chosen from the set of three submitted not because they provide better scores, but because their source code and the knowledge about their original implementation can be easily accessed. As a result of this we expect to be able to significantly improve them in the future.

The performance of the two systems is compared below for each answer type separately, as the answer extraction methodology differs from one answer type to another. In the following graphs, R_1 designates the normalized percentage of CORRECT answers, U_1 of UNSUPPORTED answers, W_1 of WRONG answers, X_1 of INEXACT answers, and Z_1 of unevaluated answers. Different colors correspond to different answer types (F=*factoid*, T=*temporally restricted*, D=*definition*, L=*list*).

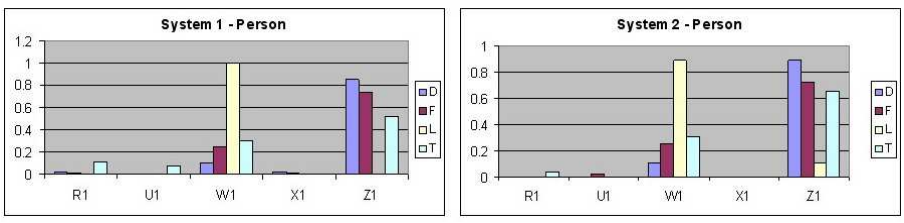

Figure 3: System comparison for the PERSON answer type

In the case of the PERSON answer type, System 1 has correctly answered a number of *factoid* and *temporally restricted* questions, while System 2 only detected correct answers for a small number of *temporally restricted* questions. There are many cases when the correct answer is retrieved, but not returned as the first answer. This is case of question 13 (where the gold answer *Melvyn Percy* appears ranked third and fifth), and question 80 the gold answer *Anthony Busuttil* appears ranked fifth. The improvement of both answer extraction modules and of the NER module employed at pre-processing need to be seriously addressed for a future CLEF participation.

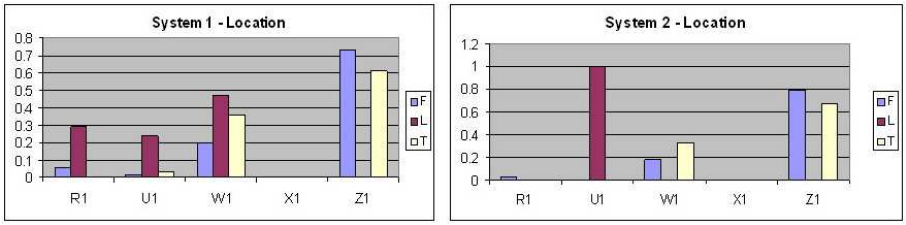

Figure 4: System comparison for the LOCATION answer type

System 1 also performs better than System 2 for questions with the LOCATION answer type. Therefore, we need to ensure that our answer extractor preserves the answer format as it appears in the related snippet. The best results have been obtained for the *list* questions asking about LOCATIONs (30% of the LOCATION *list* questions were correctly answered by System 1, while the LOCATION *list* questions were evaluated as UNSUPPORTED in the case of System 2). For a number of questions asking for LOCATIONs, due to the snippet pre-processing stage, answers detected correctly by System 2 are marked as UNSUPPORTED because the indicated snippet contains an _" instead of a whitespace.

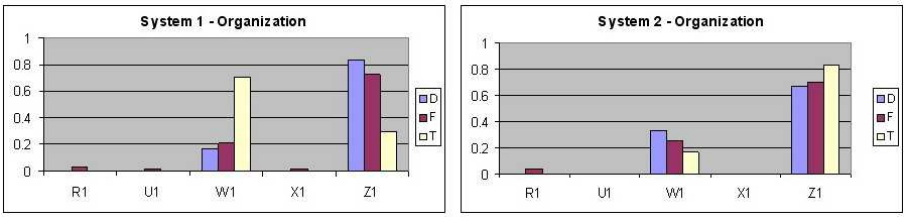

Figure 5: System comparison for the ORGANIZATION answer type

For questions asking about ORGANIZATIONs, System 2 achieves better results than System 1, but still quite low. Both systems fail, mainly because our NE Recognizer does not identify Named Entities of the type ORGANIZATION. Therefore, when our search engine looks for an entity of this type, it is obviously impossible for it to retrieve any snippets containing such entities.

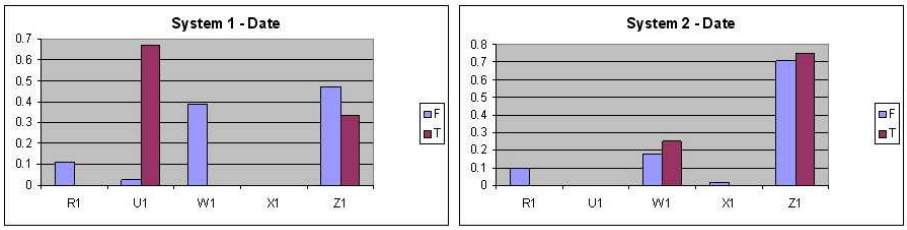

Figure 6: System comparison for the DATE expected answer type

By analyzing Figure 6, we deduce that the questions asking about a DATE and being at the same time *temporally restricted* were the most difficult questions to process. Both systems managed

to find correct answers only for the *factoid* DATE questions, but no correct answer was found for *temporally restricted* DATE questions. In the case of System 1, most of the *temporally restricted* DATE questions were considered UNSUPPORTED by the provided snippet. Also, we can see a large number of UNSUPPORTED answers for System 1, possibly due to truncating the extracted paragraphs to comply with the 500-byte limit, and leaving the answer in the truncated part. In any case, the improvement of the answer extractor is required with respect to the identification of the correct extent of the required answer. System 2 answered correctly less *factoid* DATE questions in comparison with System 1. In the future, we intend to perform a temporal pre-annotation of the corpus in order to facilitate the search for temporal expressions and to be able to provide an answer at the granularity required by the question.

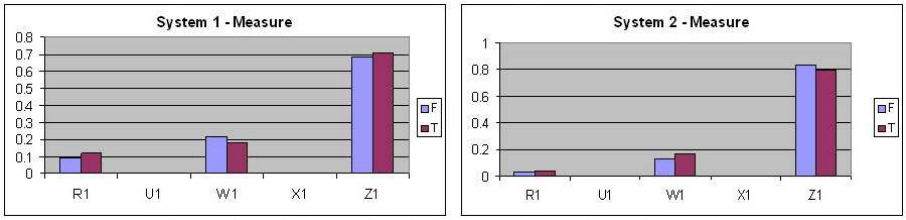

Figure 7: System comparison for the MEASURE answer type

For the MEASURE answer type, our answer extractors mainly fail due to errors in the annotation introduced at the pre-processing stage. For a number of questions, we correctly select the snippets (the gold answers are in these snippets), but we give wrong answers due to wrong annotation.

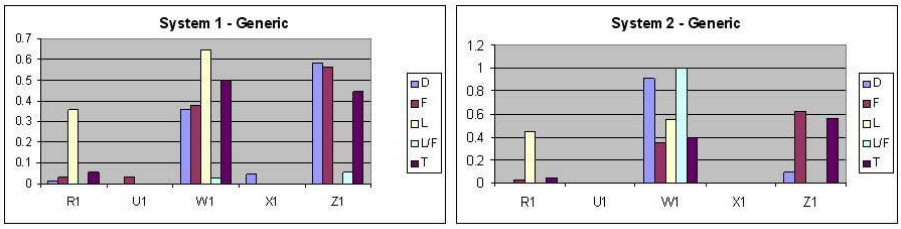

Figure 8: System comparison for GENERIC questions

All the questions that didn't fit in the previous categories (PERSON, LOCATION, ORGANIZATION, DATE nor MEASURE) are classified as GENERIC. For the GENERIC answer type, both systems achieve good results for *list* questions, but, in the case of *factoid* questions, most of the returned answers are wrong. Figure 8 shows that System 1 has significantly more correct *list* and *definition* answers, unlike System 2.

As a general overview, we may say that System 1 outperforms System 2 for most answer types. For the next competition, we will try to exploit the beneficial features of each system, at the same time improving each of the developed modules for an increased performance.

5 Conclusions

This paper describes the development stages of our cross-lingual Romanian-English QA system, as well as our participation in the QA@CLEF campaign. We have developed a basic QA system that is able to retrieve answers to Romanian questions in a collection of English documents.

Adhering to the generic QA system architecture, our system implements the three essential stages (question analysis, information retrieval and answer extraction), as well as a cross-lingual QA specific module which translates the relevant question terms from Romanian into English.

Our participation in the QA@CLEF competition included three runs. Two runs were obtained by analyzing the questions, translating the keywords, retrieving relevant passages and finding the final answers using two different answer extractors. Our third run, which unfortunately was not evaluated, was provided by the DIOGENE QA system on the basis of the output given by our Question Analysis and Term Translation modules.

The results are poor for both evaluated runs, but we declare ourselves satisfied with the fact that we have managed to develop a fully functional cross-lingual QA system and that we have learned several important lessons for our future participations.

A detailed result analysis has revealed a number of major system improvement directions. The term translation module, as a key stage for the performance of any cross-lingual QA system, is our first improvement target. However, the answer extraction module is the one we will dedicate most of our attention in order to increase its accuracy. An improved answer ranking method for the candidate answers will also form part of our priorities.

The Romanian team's debut in the CLEF competition has enriched us with the experience of developing the first Romanian-English cross-lingual QA system, at the same time setting the scene for subsequent CLEF participations.

6 Acknowledgements

The authors would like to thank the following members of the UAIC team: Cornel Bârnă, Corina Forăscu, Maria Husarciuc, Mădălina Ioniţă, Ana Masalagiu, Gabriela Mogoş, Gabriel Negară, Marius Răschip, for their help and support at different stages of system development.

Special acknowledgements are addressed to Milen Kouylekov and Bernardo Magnini for their offer to process the output of our question processor and term translation module with the IR and answer extraction modules included in DIOGENE, and for providing us with the DIOGENE run.

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] M. Kouylekov, B. Magnini, M. Negri, and H. Tanev. ITC-irst at TREC-2003: the DIOGENE QA system. In *Proceedings of the Twelvth Text Retrieval Conference (TREC-12)*, 2003.

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

[5] V. Pekar, M. Krkoska, and S. Staab. Feature weighting for cooccurrence-based classification of words. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING-04)*, 2004.

[6] G. Puscasu. A Framework for Temporal Resolution. In *Proceedings of the 4th Conference on Language Resources and Evaluation (LREC2004)*, 2004.

[7] QA@CLEF. http://clef-qa.itc.it/CLEF-2006.html, 2006.

[8] D. Tufis. Tagging with Combined Language Models and Large Tagsets. In *Proceedings of the TELRI International Seminar on "Text Corpora and Multilingual Lexicography"*, 1999.

[9] D. Tufis, D. Cristea, and S. 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.