RACAI's QA System at the Romanian-Romanian Multiple Language Question Answering (QA@CLEF2008) Main Task

Radu Ion, Dan Ştefănescu, Alexandru Ceauşu, Dan Tufiş

Research Institute for Artificial Intelligence, Romanian Academy
13, Calea 13 Septembrie, Bucharest 050711, Romania
radu@racai.ro, danstef@racai.ro, aceausu@racai.ro, tufis@racai.ro

This paper describes the participation of the Research Institute for Artificial Intelligence, Romanian Academy (RACAI) to the Multiple Language Question Answering Main Task at the CLEF 2008 competition. We present our Question Answering system answering Romanian questions from Romanian Wikipedia documents focusing on the implementation details. The presentation will also emphasize the fact that question analysis, snippet selection and ranking provide a useful basis of any answer extraction mechanism.

ACM Categories and Subject Descriptors: H.3.1 Content Analysis and Indexing – *Indexing Methods*, H.3.1 Content Analysis and Indexing – *Linguistic Processing*, H.3.3 Information Search and Retrieval – *Query formulation*, H.3.3 Information Search and Retrieval – *Search process*, H.3.3 Information Search and Retrieval – *Selection process*, I.2.7 Natural Language Processing – *Text analysis*

Keywords: Question Answering, query formulation, search engine, snippet selection, snippet ranking, question analysis, answer extraction, lexical chains.

1. Introduction

The Research Institute for Artificial Intelligence of the Romanian Academy (RACAI) is at the 3rd participation in the CLEF series of Question Answering competitions. This year (as in the previous one) we have focused on automatically answering questions in Romanian by searching their answers in Romanian Wikipedia documents. The classical requirement of the QA task is that the system must provide the shortest, syntactically well-formed string that completely answers the user's natural language question obeying the constraint that the string must be supported (contained) by a relevant text snippet which belongs to a document in the document collection. However, in the last two years, a new level of difficulty was added constraining the QA systems to resolve referential expressions and/or pronouns between questions topically grouped in a cluster in order to provide the answers.

Our *present* system is based on the one that we have developed for the previous CLEF competition (Tufiş et al., 2008c). The main differences reside in an improved query formulation module and a completely redesigned answer extraction module which uses the results of a snippet selection and ranking component which did not exist in the 2007 version of the system. We have introduced this component because it is our belief that by incrementally improving every module of the system starting with the question analysis, we are able to eliminate cascading errors which obviously affect the most critical module which is the answer extractor. Thus, our current architecture consists of the following:

1. **question analysis** in which the topic/focus articulation, question type and answer type are identified;
2. **query formulation** in which the translation from natural language question to the search engine syntactically well-formed query takes place;
3. **information retrieval** in which, using a **search engine**, the top *K* documents matching the query from the previous step are returned;
4. **snippet selection and ranking** in which, from the results of the search engine, the top *M* text snippets of a given word count are selected and ordered as to the likeliness to contain the correct answer;
5. **answer extraction** in which, using the sorted list of *M* snippets, answers candidates are extracted as syntactically well-formed substrings and returned as a ordered list in the decreasing order of the likeliness to be the correct answer to the user's question.

These modules are pipelined from 1 to 5 so that the whole QA system receives as input a natural language question and outputs an ordered list of answers along with answer confidence and support snippet. It is our intention to implement all of these modules as web services in the Semantic Web acceptance of the term (currently only query formulation and information retrieval modules are web services) so as to be able to **a)** develop each one independently of the others, **b)** easily measure the Mean Reciprocal Rank of the answer

extraction module when modifications were made to the other dependencies and **c)** easily deploy the QA system itself as a web service and with it, a QA web application.

In what follows, we will briefly describe each of our QA system modules, concluding with the presentation of our results in the QA@CLEF2008 Romanian-Romanian Wikipedia QA competition.

2. The test set of the Romanian-Romanian QA@CLEF2008 track

Following the initiative from last year, this year's test set comprised of 200 Romanian questions that were grouped in 119 topical clusters. Each cluster consisted of related questions out of which, the first one was always personal and demonstrative pronoun free. However, subsequent questions could contain pronominal references and/or referential expressions to either one of the previous questions' topic or answer. Consider the following examples from the test set:

```
(2.1 ro) 0028/3018  Cine a publicat în 1801 lucrarea Disquisitiones Arithmeticae?
(2.2 ro) 0030/3018  Câte scrieri a publicat el?
(2.1 en) 0028/3018  Who published Disquisitiones Arithmeticae in 1801?
(2.2 en) 0028/3018  How many writings did he publish?
```

The question number 0028 is the first question in the group 3018 and it asks for the author of the "Disquisitiones Arithmeticae". The third question of the group (the second in our example), inquires about the number of writings of the author of "Disquisitiones Arithmeticae". This is an example where pronoun "he" refers back to the answer of a previous question. Another example is where a referential expression of a subsequent question refers back to the topic of a previous question:

```
(2.3 ro) 0025/3016  Pe ce continent este situată Bulgaria?
(2.4 ro) 0026/3016  Când şi-a recâştigat această ţară independenţa completă?
(2.3 en) 0025/3016  On which continent is Bulgaria situated?
(2.4 en) 0026/3016  When did this country regain its complete independence?
```

Here, the second question (no 0026) asks for the date in which the country "Bulgaria", the topic of the first question, regained its complete independence.

Searching for the motivation that led to this added level of difficulty when analyzing the questions, we discover that the organizers of the QA@CLEF track (Giampiccolo et al., 2007) took after the TREC series QA competitions (http://trec.nist.gov/) where the systems were also given clusters of topically related questions containing pronominal/referential expressions. We checked the TREC QA 2007 data for English (http://trec.nist.gov/data/qa/t2007_qadata.html) and we have found that there is one important difference: whereas CLEF question analysis requires full anaphora resolution, the TREC "anaphora resolution" merely entails the replacement of the pronoun/referential expression with the topic of the cluster which, and this is important, *is given* for every cluster. Furthermore, there is no such thing as searching for the referent in previous questions of the cluster or in their answers: the referent is always bound to the given topic of the cluster.

It is our belief that this added level of difficulty to the question analysis in QA@CLEF is counterproductive. If answering a question requires that you correctly answered *a previous one* (in the case of pronominal reference to a previous answer), the whole enterprise misses the point: the true ability of a QA system to answer natural language questions of a user which is mainly interested in finding the desired information and therefore, presumably, not inclined to "fool" the system by asking "complicated" questions (Tufiş and Popescu, 1991). On the other hand, it seems that the basic problem, answering unambiguous, simple questions, is far from being solved at least for Romanian (and for other languages as well considering that Giampiccolo et al. (2007) report an average accuracy not exceeding 30%) so introducing additional levels of difficulty does not seem appropriate at present.

In the light of these reflections, and recognizing the fact that our system does not deal at all with anaphora because it is a kind of processing at question analysis level we think the average QA user will almost never need, we have devised a second test set (from hereon called "the normalized test set" as opposed to "the official test set") in which we have manually identified and replaced all referents in all referential expressions/pronouns using the official Romanian-Romanian QA@CLEF2008 Gold Standard of Answers when we dealt with answers referents. We thus wanted to know how our QA system performs on the classical QA task in which the question itself supplies sufficient information so as to be able to identify the answer to it. Following are the examples above in which we have done the replacements:

```
(2.1 ro) 0028/3018  Cine a publicat în 1801 lucrarea Disquisitiones Arithmeticae?
(2.2 ro) 0030/3018  Câte scrieri a publicat Carl Friedrich Gauss?
(2.3 ro) 0025/3016  Pe ce continent este situată Bulgaria?
(2.4 ro) 0026/3016  Când şi-a recâştigat Bulgaria independenţa completă?
```

We will continue with our QA system components presentation and we will conclude by presenting the performance of the system using both the official test set and the normalized test set. We should note that the following presentation follows closely but extends significantly the one in (Ion et al., 2008).

3. The search engine

The document collection remained unchanged from the 2007 edition of the QA@CLEF. This collection is composed of 43486 Romanian language documents from Wikipedia (http://ro.wikipedia.org/). Each document has a title and several sections made up from paragraphs. All the logical sections of the documents were preprocessed with the TTL module (Ion, 2007) to obtain POS tagging, lemmatization and chunking of the text within.

The search engine is a C# port of the Apache Lucene (http://lucene.apache.org/) full-text searching engine. Lucene is a Java-based open source toolkit for text indexing and Boolean searching allowing queries formed with the usual Boolean operators such as AND, OR and NOT. Furthermore, it is capable to search for phrases (terms separated by spaces and enclosed in double quotes) and also to allow boosting for certain terms (the importance of a term is increased with the caret character '^' followed by an integer specifying the boost factor). We also used the field-specific term designation: a term may be prefixed by the field name to constrain the search to specific fields (such as title or text for instance) in the document index.

For the construction of the index, we considered that every document and every section within a document have different fields for the surface form of the words and their corresponding lemmas. This kind of distinction applies to titles and paragraph text resulting in four different index fields: title word form (**`title`**), title lemma (**`ltitle`**), paragraph word form (**`text`**) and paragraph lemma (**`ltext`**). We used the sentence and chunk annotation (from the TTL output) to insert phrase boundaries into our term index: a phrase query cannot match across different chunks or sentences. Thus, for instance, if we want to retrieve all documents about the TV series Twin Peaks, we would first like to search for the phrase "Twin Peaks" in the title field of the index (Lucene syntax **`ltitle:`**`"Twin Peaks"`) and then, to increase our chance of obtaining some hits, to search in the word form field of the index for the same phrase (Lucene syntax **`text:`**`"Twin Peaks"`). Consequently, this Lucene query would look like this: **`ltitle:`**`"Twin Peaks"` **`OR text:`**`"Twin Peaks"`.

The result of searching is a list of documents which is ordered by the relevance score of each document with respect to the posed query. Within each document, if a condition applies (see below), its sections are also ranked and listed in decreasing order with respect to the relevance to the query. We have used Lucene's default scoring formulas when ranking documents and their sections.

Document retrieval is done with a special technique that allows for automatic query relaxation. Given a Boolean query that is a conjunction of n terms, the system will first try to match all of the query terms (which may be prefixed by different fields) against the document index. If the search is unsuccessful, the system will try to exhaustively match $n - k$ ($1 \leq k < n$) of the query terms until a relaxed query (which now contains $m < n$ terms) returns at least one document from the document index. If the query is not a conjunction of terms (has a more complicated structure involving grouping and usage of other Boolean operators) then it is submitted as is and the results are returned to the application. It is the application's responsibility to deal with too specific queries that have a void result.

If the query is a conjunction of terms, the search engine also provides section ranking within each of the returned documents. This is achieved by automatically performing a second search within these documents using the section index and a new query. The new query is composed by joining the terms from the relaxed query that produced the document list with the disjunction operator (OR).

The Romanian Wikipedia document retrieval mechanism is available as a web service (Tufiş et al., 2008b). The WSDL description can be found at http://nlp.racai.ro/webservices/SearchRoWikiWebService.asmx?WSDL and a web application that uses this web service is located at http://nlp.racai.ro/webservices/SearchRoWiki.aspx . The web application features a search box in which, for a given Lucene query, the list of top K ranked documents is displayed in the HTML format (see Figure 1). The web service itself returns an XML document containing the list of the first K documents. This XML document is the input of the snippet selection and ranking module.

Figure 1: The output of the search web application for the question "*Cine este Ares?/Who is Ares?*" with the Lucene query "**ltitle:**Ares **ltext:**Ares **title:**ares **text:**ares"

4. Question analysis and query formulation

The question analysis produces the focus and the topic of the question and was described in (Tufiş et al., 2008c). Basically, it uses the linkage of the question obtained with LexPar (Ion, 2007) to identify linking patterns that describe the syntactic configuration of the focus, the main verb and the topic of the question. For instance, in the question "*Câte zile avea aprilie înainte de 700 î.Hr.?*"/"*How many days did April have before 700 B.C.?*" the focus is the word "*zile*"/"*days*" and the topic is "*aprilie*"/"*April*" because the linkage of this question contains the links interrogative determiner "*Câte*"/"*How many*" – noun "*zile*"/"*days*", noun "*zile*"/"*days*" – main verb "*avea*"/"*did have*" and main verb "*avea*"/"*did have*" – noun "*aprilie*"/"*April*" which determine a syntactic pattern in which the focus is the first noun and the topic, the second noun.

The query formulation strategy improves the one described in (Tufiş et al., 2008c) which was successfully used in the Romanian-Romanian QA@CLEF2007 track. The input question must first be preprocessed with the TTL module to obtain POS tagging, lemmatization and chunking information.

Query formulation from a POS tagged, lemmatized and chunked input question basically constructs a conjunction of terms which are extracted from the chunking information of the input sentence. Specifically, the algorithm considers all the noun phrases of the question from which it constructs terms. Each term is prefixed by a text or title field and is present both in lemma and word form.

The CLEF 2007 version of the algorithm used to take into account all the word boundary substrings of each noun phrase regardless of their likeliness to appear. For instance, for the noun phrase "*cele mai avansate tehnologii ale armatei americane*"/"*the most advanced technologies of the US army*", terms like "*mai avansate tehnologii ale*" or "*cele mai avansate*" were valid. The present version of the question formulation algorithm fixes this aberration by constraining the substrings to be proper noun phrases themselves. In addition to that, the assignment of fields to each term was revised. Following, is the summary of modifications:

1. substring starting or ending with words of certain parts of speech are not considered terms; for instance substrings ending with adjectives or articles or beginning with adverbs;
2. substrings that do not contain a noun, a numeral or an abbreviation are not considered terms;
3. substrings starting with words other than nouns, numerals or abbreviations are not to be searched in the title field;
4. single word terms in occurrence form are not to be searched in the title field.

The improvement of the query formulation algorithm is exemplified for the question "*Cine erau părinţii lui Ares, conform mitologiei greceşti?*"/"*Who were Ares' parents, according to the Greek mythology?*". The older version of the query formulation produced the query (the space between terms signifies the existence of the AND operator)

ltitle:"părinte lui Ares" **ltext:**"părinte lui Ares" **ltext:**"lui Ares" **ltext:**"părinte lui" **ltitle:**"mitologie grecesc" **ltext:**"mitologie grecesc" **ltitle:**mitologie **ltext:**mitologie **ltext:**grecesc **ltitle:**părinte **ltext:**părinte **ltitle:**Ares **ltext:**Ares **title:**"părinţii lui ares" **text:**"părinţii lui ares" **text:**"lui Ares" **text:**"părinţii lui" **title:**"mitologiei greceşti" **text:**"mitologiei greceşti" **title:**mitologiei **text:**mitologiei **text:**greceşti **title:**părinţii **text:**părinţii **title:**ares **text:**ares

while the new version of the query after applying steps 1 – 4 is the following

ltitle:"părinte lui Ares" **ltext:**"părinte lui Ares" **ltitle:**"mitologie grecesc" **ltext:**"mitologie grecesc" **ltitle:**părinte **ltext:**părinte **ltitle:**Ares **ltext:**Ares **ltitle:**mitologie **ltext:**mitologie **title:**"părinţii lui ares" **text:**"părinţii lui ares" **title:**"mitologiei greceşti" **text:**"mitologiei greceşti" **text:**părinţii **text:**ares **text:**mitologiei

Thus, we can see that from a query of 26 terms, we ended up with a smaller query of 17 terms. From the 26 initial terms, improper and unlikely to appear terms such as **ltext:**"lui Ares", **ltext:**"părinte lui", **text:**"lui Ares", **text:**"părinţii lui" and so on have been eliminated.

The importance of the single word heuristic for title searching (step 4) becomes clearer in the light of a question like "*Ce este Pământul?*"/ "*What is the Earth?*". The queries generated by the two versions (older (a) and present (b)) are: (a) **ltitle:**pământ **ltext:**pământ **title:**pământul **text:**pământul and (b) **ltitle:**pământ **ltext:**pământ **text:**pământul. The older query retrieved as the top document a document with the title "*Pământul de Mijloc*"/"*Middle Earth*" referring to a fictional land created by J.R.R. Tolkien because of the existence of the term **title:**pământul and because of the "match all if you can" heuristic of the search engine which matched both the word form and the lemma of the term in the title field. With the second query, all the terms were matched as in the previous case but the top scored document now refers to the planet Earth (Middle Earth document being listed below probably because its title is longer than the title of the planet Earth document which is "*Pământ*").

We evaluated the retrieval accuracy of this query formulation algorithm onto *the normalized question* test set of the Romanian-Romanian QA@CLEF2008 track in which we participated this year. We used this test set in order to give a fair chance to the algorithm to discover the relevant terms that would produce the expected documents.

Table 1: Query formulation algorithm improvements onto the normalized test set

	MRR	Coverage
Initial	~0.7000	0.7500
Step 1	0.7366	0.7624
Step 2	0.7455	0.7715
Step 3	0.7689	0.8012
Step 4	0.7776	0.8092

The query structure (its terms in our case) directly influences the *accuracy* and the *coverage* of the search engine. The accuracy is computed as a *Mean Reciprocal Rank* (MRR) score for documents (Voorhees, 1999), while the coverage is practically the recall score (coverage is the upper bound for the MRR). Although we primarily aim for covering all the questions (which means that we want to relax the queries in order to get documents/sections containing answers for as many questions as possible), a good MRR will ensure that these documents/sections will be among the top returned. Consequently, the detection of the exact answer should be facilitated if this procedure considers the ranks assigned by the search engine to the returned documents. The greater the MRR score, the better the improvement.

As Table 1 shows, starting from a MRR of around 0.7 and a coverage of 0.75 obtained with the 2007 version of the query formulation algorithm, the improved query formulation algorithms now achieves a MRR of 0.7776 and a coverage of 0.8092. The figures were computed using the reference Gold Standard of the Romanian-Romanian QA@CLEF2008 track in which for each question, the document identifier of the document containing the right answer is listed. We have not considered the questions which had a NIL answer and as such, no document identifier assigned.

The implementation of this query formulation algorithm is a web service (the WSDL description of which can be found at http://shadow.racai.ro/QADWebService/Service.asmx?WSDL) that takes the Romanian question as input and returns the query. To obtain POS tagging, lemma and chunking information, the web service uses another web service, TTL (Tufiş et al, 2008b).

5. Snippet selection and ranking

The snippet (or passage) selection is important to the answer extraction procedure because the answers should be sought only into small fragments of text that are relevant with respect to the query and implicitly to the question. This is because, usually, the answer extraction procedure is computationally expensive (different reasoning mechanisms, parsing, ontology interrogation, etc. are involved) and thus, would be best to apply it onto small fragments of text.

Snippet selection uses the question analysis to identify passages of text that, potentially, contain the answer to the question. For each section of each returned document, the snippet selection algorithm considers windows of at most *N* words at sentence boundary (that is, no window may have fewer than *N* words but it may have more than *N* words to enforce the sentence boundary condition). Each window is scored as follows (each word is searched in its lemma form):

- if the focus of the question is present, add 1;
- if the topic of the question is present, add 1;
- if both the topic and the focus of the question are present, add 10;
- if the *k*-th dependant of the focus/topic of the question is present add 1 / (*k* + 1). The *k*-th dependent of a word *a* in the linkage of a sentence is the word *b*, if there exists a path of length *k* between *a* and *b*.

The above heuristics are simple and intuitive. Each window in which either focus or topic are found, receives one point. If both are to be found, a 10 point bonus is added because the window may contain the reformulation of the question into a statement which will thus resolve the value of the focus. The last heuristic aims at increasing the score of a window which contains dependents of the focus and/or topic but with a value which decreases with the distance (in links) between the two words in order to penalize snippets that contain long distance dependents of the focus/topic that may be irrelevant. The selection algorithm will retrieve at most *M* top-scored snippets from the documents returned by the search engine. A snippet may be added only if its selection score is greater than 0.

The snippet selection algorithm provides an initial ranking of the snippets. However, there are cases when the focus/topic is not present in its literal form but in a semantically related form like a synonym, hypernym, etc. This problem is known as the "lexical gap" between the question formulation and the text materialization of the answer. In these cases, our snippet selection procedure will assign lower scores to some of the important snippets because it will not find the literal representation of the words it looks for. To lighten the impact of this problem onto the snippets' scores, we developed a second ranking method which uses *lexical chains* to score the semantic relatedness of two different words.

5.1. Lexical chains

The term "lexical chain" refers to a set of words which, in a given context (sentence, paragraph, section and so on), are semantically related to each other and are all bound to a specific topic. For instance, words like "*tennis*", "*ball*", "*net*", "*racket*", "*court*" all may form a lexical chain if it happens that a paragraph in a text contains all of them. Moldovan and Novischi (2002) used an extended version of the Princeton WordNet (http://wordnet.princeton.edu/) to derive lexical chains between the meanings of two words by finding semantic relation paths in the WordNet hierarchy. Thus, a lexical chain is not simply a set of topically related words but becomes a path of meanings in the WordNet hierarchy.

Following (Moldovan and Novischi, 2002) we have developed a lexical chaining algorithm using the Romanian WordNet (RoWN) (Tufiş et al, 2008a) that for two words in lemma form along with their POS tags returns a list of lexical chains that exist between the meanings of the words in the Romanian WordNet. Each lexical chain is scored as to the type of semantic relations it contains. For instance, the synonymy relation receives a score of 1 and a hypernymy/hyponymy relation, a score of 0.85. Intuitively, if two words are synonymous, then their semantic similarity measure should have the highest value. The score of a lexical chain is obtained by summing the scores of the semantic relations that define it and dividing the sum to the number of semantic relations in the chain. All the semantic relations present in Romanian WordNet have been assigned scores (inspired by those in (Moldovan and Novischi, 2002)) between 0 and 1. Thus, the final score of a lexical chain may not exceed 1.

Basically, the lexical chaining algorithm expands the semantic frontier of each end of the lexical chain and searches for an intersection of frontiers. When such an intersection is found (more than one is found actually) it stops and retrieves each chain by backtracking from the intersection point to the source and the target word. The semantic frontier of a literal paired with a meaning identifier (let's call it $l(a)$) is a set of $l(x)$ which consist of:

- all the synset members of $l(a)$ except $l(a)$;
- all $l(i)$ in all the synsets that form a direct paradigmatic relation (such as hypernymy, meronymy, etc.) with the synset of $l(a)$;

- every content word from the gloss of *l*(*a*) paired with the indefinite meaning "?";

Initially, the procedure receives two POS disambiguated, Romanian lemmas along with the indefinite meaning "?". When encountering the indefinite meaning, the frontier expansion will generate all the synsets containing that literal. The expansion keeps track of the nodes that have already been expanded so that they won't be expanded again. Next to the already established semantic relations in RoWN we have added two more: the synonymy (**ss**) relation that is established between two members of a synset and the gloss (**gloss**) relation that holds between members of a synset and the content words of the gloss (glosses are POS tagged and lemmatized using TTL).

Our lexical chaining algorithm differs from the one in (Moldovan and Novischi, 2002) in several aspects. Firstly, RoWN does not have word sense disambiguated glosses and as such, we had to consider all the senses of a gloss literal (we have to assign the gloss literal the indefinite meaning "?"). Secondly, our algorithm expands the semantic frontier of a meaning more than once (it has a parameter which has been experimentally set to 3) allowing for discovery of "deeper" lexical chains. In practice we have observed that finding a lexical chain between two words with a depth cut to 3 relations may take an amount of time anywhere between 50ms to 30s. But, experimentally we have found that most correct lexical chains are found in an amount of time less than 1s and thus we have imposed 1s of running time to finding lexical chains between two words.

Here are the 3 of 7 lexical chains found between nouns *fiu*(?)/*son(?)* and *tată*(*?*)/*father*(?) with 1s of running time:

```
(5.1.1) 0.333
     fiu(?) near_antonym fiică(1) <=> fiică(1) near_antonym copil(?) gloss tată(?)

(5.1.2) 0.666
     fiu(?) hypernym băiat(1) <=> băiat(1) hypernym copil(?) gloss tată(?)

(5.1.3) 0.733
     fiu(?) ss băiat(5) <=> băiat(5) ss copil(?) gloss tată(?)
```

The "**<=>**" markup signifies the frontier intersect point. From that point, the chain is reconstructed by backtracking left to the source of the lexical chain and right to the target of the lexical chain. For instance, example (5.1.1) may be read as follows: there is a synset which contains the literal "*fiu*"/"*son*" and that is a **near_antonym** of another synset which contains the literal "*fiică*"/"*daughter*" with the sense identifier 5. On the other hand, the latter synset is a **near_antonym** with a synset containing the literal "*copil*"/"*child*" which appears in the **gloss** of a synset containing the literal "*tată*"/"*father*". It is obvious that the question marks standing for the meaning identifiers can easily be recovered. But what we are after here primarily is the score of the lexical chain. The higher it is, the stronger the semantic relatedness between the two words should be. Again, in example (5.1.1), the **near_antonym** relation has an experimental score of 0.4 and the **gloss** relation, 0.2. Thus the score of this lexical chain is (0.4 + 0.4 + 0.2) / 3 = 0.333.

5.2. Evaluating the snippet selection and ranking

Using the lexical chaining mechanism, we were able to re-rank the snippets that were selected with the previous procedure by computing the best lexical chain scores between focus, topic and their dependents and the words of the window. We have not computed lexical chains for words that were identical because this case is successfully covered by the snippet selection mechanism. Thus, for instance, for the question "*Câți oameni încap în Biserica Neagră din Brașov?*"/"*How many people fit into the Black Church in Brașov?*" the text snippet "*Biserica Neagră este cel mai mare edificiu de cult în stil gotic din sud-estul Europei, măsurând 89 de metri lungime şi 38 de metri lățime. În această biserică încap circa 5.000 de persoane.*" was able to receive a higher score due to the fact that the question focus "*oameni*" is materialized as "*persoane*" (they are synonymous).

As with the queries, we wanted to evaluate the two methods of snippet ranking individually and in combination using the normalized test set. We have set *N* (the number of words in a snippet) to 10 and 50 (these settings for *N* roughly correspond to 50-byte and 250-byte runs of the previous TREC QA competitions) and *M* (the number of retrieved snippets) to 20. We have also considered only the top 10 documents returned by the search engine. We counted a snippet (MRR style) only if it contains the answer *exactly* as it appears in the official Romanian-Romanian QA@CLEF2008 Gold Standard of Answers (no interest in NILs). Table 2 summarizes the results.

Table 2: MRR performance of the snippet selection and ranking algorithm on the normalized test set

N	Key word ranking	Lexical chain ranking	The combination	Coverage
10	**0.4372**	0.3371	0.4037	0.6894
50	0.4589	0.3679	**0.4747**	0.7

The combination of the two ranking methods consists in simply adding the scores provided for each snippet. When the snippet contains 10 words, the lexical chaining ranking method does not help the keyword ranking method because the semantic relatedness evidence is reduced by the short size of the snippet. When the snippet size increases (50 words), the contribution of the lexical chaining is more significant and this is reflected in the MRR score.

Unfortunately, the snippet selection and ranking module was developed and tested after the Romanian-Romanian QA@CLEF2008 track has ended. At the time of the writing, we didn't test this module onto the official test set but we are able to provide snippet MRR calculation from our official results where snippets were directly provided by the answer extraction procedure. Since only the first three answers were taken into consideration, it means that *M* equals 3 in this case. We count (MRR style) all the "YES" judgments from the `<answer-snippet>` element of the XML result file. This gives us a MRR of 0.2533 (coverage 0.325) for the first run and a MRR of 0.3041 (coverage 0.38) for the second run. All these figures are significantly lower than our current figures, using the normalized test set.

6. The answer extraction procedure

Answer extraction is responsible of extracting that syntactically well-formed substring that completely answers the user's question. Our answer extraction module relies on the question analysis and on the lexical chains algorithm. Since we cannot employ the services of a Named Entity Recognition module because we do not have one for Romanian, the answer type information coming from the question analysis cannot be used. The present version of the answer extraction algorithm operates on the snippets provided by the snippet selection and ranking algorithm. It performs the following steps:

1. if the question has a partially instantiated focus (like in a WH-determiner followed by a noun or an imperative verb followed by its complement) then, for each snippet in the list, compute the maximal lexical chain score between focus and each noun in the snippet selecting the noun phrase in which the noun appears as a candidate answer. Order the list of candidate answers using a key given by a sum containing the following terms: lexical chain score, snippet ranking and section/document ranking containing the snippet;
2. if the question focus is a WH-pronoun/adverb, then, for each snippet in the list, compute the maximal lexical chain score between question main verb and each verb in the snippet and extract as candidate answers, substrings before and after the snippet verb at clause or sentence boundary. As in the previous step, order the list of candidate answers using the same key.
3. candidate answers that have a higher lexical-chain-like affinity with the topic of the question and its dependents are penalized (we want to avoid repeating what was said in the question).

To exemplify the operation of this very simple algorithm, consider the question "*Câte **zile** avea aprilie înainte de 700 î.Hr.?*"/"*How many **days** did April have before 700 B.C.?*" and the text snippet:

```
Înainte/înainte/Rgp/Ap#1                          0
de/de/Spsa/Pp#1                                   0
anul/an/Ncmsry/Pp#1,Np#1                          1
700/700/Mc/Pp#1,Np#1                              0
î.Hr./î.Hr./Yr/Pp#1,Np#1                          0
,/,/COMMA/                                        0
luna/lună/Ncfsry/Np#2                             0.9
aprilie/aprilie/Ncms-n/Np#2                       0.733
era/fi/Vmii3s/Vp#1                                0
a/al/Tsfs/Np#3                                    0
doua/doi/Mofs-l/Np#3                              0
lună/lună/Ncfsrn/Np#3                             0.9
a/al/Tsfs/Np#3                                    0
anului/an/Ncmsoy/Np#3                             1
în/în/Spsa/Pp#1                                   0
calendarul/calendar/Ncmsry/Pp#1,Np#4              0.733
roman/roman/Afpms-n/Pp#1,Np#4,Ap#1                0
şi/şi/Crssp/                                      0
avea/avea/Vmii3s/Vp#2                             0
29/29/Mc/Np#5                                     0
de/de/Spsa/Pp#2                                   0
zile/zi/Ncfp-n/Pp#2,Np#5                          1
././PERIOD/                                       0
```

The left column contains the lemma, POS tagging and chunking analysis of the text snippet and the right column shows the lexical chains maximal scores between the focus of the question "*zi*"/"*day*" and the nouns of the snippet. Thus, competing for being the right answer to the user's questions would be the strings "*anul 700 î.Hr.*" (noun phrase no 1, Np#1), "*a doua lună a anului*" (Np#3) and "*29 de zile*" (Np#5) as they contain nouns that have a lexical chain score of 1 when linked with "*zi*". But the first string will be penalized because it contains words from the topic of the question. Thus, we are left with two possible answers out of which "*29 de zile*" is the right one but not using NER, we cannot boost its score.

As we have previously stated, when we submitted our results to the Romanian-Romanian QA@CLEF2008 track, we didn't have the snippet selection and ranking module and as such, our answer extraction module directly operated on the results of the search engine. At the time of writing, we did not test the answer extractor using the snippet selection and ranking module with the official test set. The following figures will be provided (Table 3 summarizes the results):

- MRR and coverage using the official test set by counting the "R" (right answers) judgments from the XML official result file looking in the `<judgment>` element;
- MRR and coverage using the official test set by counting the "X" (inexact answers) judgments from the XML official result file looking in the `<judgment>` element;
- MRR and coverage using the normalized test set and the snippet selection and ranking module by counting (MRR style) the *exact* answers found in the official Romanian-Romanian QA@CLEF2008 Gold Standard of Answers (no interest in NILs).

Table 3: The answer extraction accuracy over the two test sets

Runs	**MRR**	**Coverage**
ICIA081RORO (official test set) Right (R)	0.0683	0.095
ICIA081RORO (official test set) ineXact (X)	0.0691	0.09
ICIA082RORO (official test set) Right (R)	0.1233	0.155
ICIA082RORO (official test set) ineXact (X)	0.0633	0.08
SSR (normalized test set) Right (R)	**0.1815**	**0.365**

The official test set contains 200 questions. We have submitted two runs: the first run, ICIA081RORO, was obtained by applying the answer extraction algorithm over the first 10 documents returned by the search engine when giving the query obtained from the first question in the cluster. All subsequent questions in the same cluster were answered from these 10 documents. The second run, ICIA082RORO, was the same as the first run except that: **a)** for definition type questions we have applied System A from (Tufiş et al, 2008c) which is specialized to answering definition type questions and **b)** for QA@CLEF2008 task, System A was modified slightly to answer some factoid questions and as such, any common factoid answers between this answer extraction algorithm and the modified System A was also preferred in the output.

As the Table 3 shows, the answer extraction procedure working on the output of the snippet selection and ranking module (SSR) and also on the 200 question normalized test set performs much better than applying the same answer extraction directly onto the results of the search engine and using ambiguous questions. Of course, this is to be expected but we want to emphasize that *these kinds of results need improving* and not the ones obtained from asking ambiguous questions. Also, we have shown that the same answer procedure greatly improves if an intermediate step selecting the snippets is involved. Our immediate goal is then to push the 0.1815 MRR figure to the possible achievable maximum which is the coverage of 0.365 disregarding completely the more complex test set which is the official test set. After we achieve an acceptable, over 0.5 MRR using "normalized" questions, we can start thinking of how to "resolve" referential expressions/pronouns within a cluster of questions.

7. Conclusions

It seems that with our newly added modules to our QA system and with all the improvements, we have in fact, taken a step back from the results we have obtained in the Romanian-Romanian QA@CLEF2007 track where we have obtained a MRR score of 0.3 on the official test set. These explanations might hold: **a)** we answered all questions in a cluster by examining documents returned for the query derived only from the first question when, in fact this was the right thing to do as more often than not, all the questions in a cluster were asked from a single document, **b)** we used a different answer extractor based on the linkage of the returned documents. What we can say for sure is that the 0.3 MRR score in 2007 *was not due* to our anaphora resolver which was an ad hoc solution designed exclusively for dealing with the 2007 official test set.

In 2008, we have dropped the linkage extraction of answers because even if a linkage is a pseudo-dependency-like analysis of a sentence, it's still not a full syntactic analysis and as such cannot offer the useful information (such as subject and direct object at least) that will make the difference. We also treated each question in its own right and didn't even attempt to resolve referential expressions/pronouns between questions simply because we do not have such a module and we thought that building something on the fly just for this task will not have future value. As to the normalized version of the test set, for a fair comparison, we will have to "normalize" the 2007 test set first and then run our present QA system on it.

As we have read on the clef-qa@list.fbk.eu mailing list, there was a proposal to terminate the current QA@CLEF task mainly because there are not enough systems to base a comparison on. *We cannot stress enough the importance of this track* and the best argument we can give is that our QA system practically sprang into existence because of the QA@CLEF. QA@CLEF is, to the best of our knowledge, the forum that evaluates QA systems for many European languages and thus, terminating it would be a great loss for European QA research. One way to compare several QA systems operating on different languages is to have a parallel corpus containing all the languages and English (we also think that JRC-Acquis is a good choice). Each QA system may operate by supplying the answer in its language (Spanish, Dutch, Romanian, Bulgarian and so on) and then, via word or phrase alignment, the organizers may obtain the output in English which is ready to be compared to an English Gold Standard of Answers.

In conclusion, whatever direction the future QA@CLEF is going to take, we will welcome it as long as we will still have a QA@CLEF task to participate in.

REFERENCES

GIAMPICCOLO, D., PEÑAS, A., AYACHE, C., CRISTEA, D., FORNER, P., JIJKOUN, V., OSENOVA, P., ROCHA, P., SACALEANU, B., SUTCLIFFE, R., *Overview of the CLEF 2007 Multilingual Question Answering Track*, In Alessandro Nardi and Carol Peters (eds.), "Working Notes for the CLEF 2007 Workshop", Budapest, Hungary, 22 p., 2007.

ION, R., *Word Sense Disambiguation Methods Applied to English and Romanian*, PhD thesis, Romanian Academy, Bucharest, 2007.

ION, R., ŞTEFĂNESCU, D., CEAUŞU, A., *Important Practical Aspects of an Open-domain QA System Prototyping*, Proceedings of the Romanian Academy, Series A, 6 p., The Publishing House of the Romanian Academy, Bucharest, 2008 (submitted).

MOLDOVAN, D., NOVISCHI, A., *Lexical Chains for Question Answering*, Proceedings of COLING 2002, Taipei, Taiwan, pp 674 – 680, 2002.

TUFIŞ, D., ION, R., BOZIANU, L., CEAUŞU, AL., ŞTEFĂNESCU, D., *Romanian WordNet: Current State, New Applications and Prospects*, In Attila Tanács et al. (eds.): "Proceedings of the Fourth Global WordNet Conference (GWC 2008)", Szeged, Hungary, pp 441 – 452, 2008a.

TUFIŞ, D., ION, R., CEAUŞU, AL., ŞTEFĂNESCU, D., *RACAI's Linguistic Web Services*, Proceedings of the Sixth International Language Resources and Evaluation (LREC'08), Marrakech, Morocco, 2008b.

TUFIŞ, D., POPESCU, O., *IURES2: Natural Language Environment and The Computer Speak Paradigm*, Proceedings of the International Conference for Young Computer Scientists, Beijing, China, 1991.

TUFIŞ, D., ŞTEFĂNESCU, D., ION, R., CEAUŞU, A., *RACAI's Question Answering System at QA@CLEF2007*, In Carol Peters et al. (eds.): "Evaluation of Multilingual and Multi-modal Information Retrieval 8th Workshop of the Cross-Language Evaluation Forum, CLEF 2007, Revised Selected Papers", Lecture Notes in Computer Science, Springer-Verlag, 2008c. (in press).

VOORHEES, E.M., *The TREC-8 question answering track report*, Proceedings of the 8th Text Retrieval Conference, Gaithersburg, Maryland, USA, pp. 77-82, 1999.