Proceedings of the International Workshop on Intelligent Information Systems
IIS'2011, September 13-14, 2011, Chisinau, Republic of Moldova

Natural Language Question Answering in Open Domains

Dan Tufiş

Abstract: With the ever-growing volume of information on the web, the traditional search engines, returning hundreds or thousands of documents per query, become more and more demanding on the user patience in satisfying his/her information needs. Question Answering in Open Domains is a top research and development topic in current language technology. Unlike the standard search engines, based on the latest Information Retrieval (IR) methods, open domain question-answering systems are expected to deliver not a list of documents that might be relevant for the user's query, but a sentence or a paragraph answering the question asked in natural language. This paper reports on the construction and testing of a Question Answering (QA) system which builds on several web services developed at the Research Institute for Artificial Intelligence (ICIA/RACAI). The evaluation of the system has been independently done by the organizers of the ResPubliQA 2009 exercise and has been rated the best performing system with the highest improvement due to the natural language processing technology over a baseline state-of-the-art IR system. The system was trained on a specific corpus, but its functionality is independent on the linguistic register of the training data.

Keywords: Open Domain search, Question Answering Evaluation, question analysis, query formulation, search engine, multi-factored training, minimal error rate training, paragraph selection/ranking, lexical chains, web services, workflow.

1 Introduction

Research in Natural Language Processing (NLP) has generated high impact results for the internet global society. The most advanced internet search engines are indexing tens of billions of documents, containing thousands of bilions of words in more than 100 languages. The informational content of the virtual space is fabulous. Rightfully one may say that for any rational information request there exists at least one relevant answer in the cyberspace. However, finding the answers and

evaluating their quality are still research problems, for which the traditional approaches became insufficient. The Intelligent Information Retrieval (IRR) and Databases technologies, the pillars on which the best performing internet search engines are built, are synergetically coupled, both on the theoretical and applicative levels, with the language and speech technologies in the quest for the relevant documents and identification as precise as possible of the informational content pertaining to a search query in the cyberspace.

As frequently happens, new ideas meet opposition or suspicion, and as such, for a while the idea that a question-answering system could be a companion to an intelligent information retrieval system, both in terms of utility and performance, had few supporters (especially in the commercial world). The prevalent opinion was (and sometimes is) that the performances of a state-of-the-art IRR system could hardly be surpassed by mixing it with a question-answering system. Such a view is basically misguiding because the two types of systems solve related, but different problems. It is well known that a search engine (Google, Bing, Yahoo, etc.) answers a query, expressed by a list of keywords (possibly related by some logical operators), with an ordered list of documents that are likely to contain the needed information. Depending on the user's ability in selecting the proper keywords, the interrogation result might be an empty list or, a list containing hundreds or thousands of documents, the user being supposed to look into these documents for the required information. More often than not, the user inspects the top documents in the list and, if his/her keywords were enough selective the information need might be satisfied. Otherwise, the user may decide to reformulate the query (if (s)he knows how to do it), providing more search criteria or adding additional restrictions.

On the other hand, an open domain question-answering system (ODQAS) does not provide a list of documents that might contain the answer, but a ranked list of probable answers, extracted from the documents which were rated as most relevant. Architecturally, a search engine is just a module (essential, indeed) of a question-answering system (Figure 1). A natural language question is processed by a specialised natural language module (NL module 1) which generates a formal query for the search engine. During this pre-processing phase, the question is

tokenized, tagged, lemmatized, parsed and most significant content words are selected. The type of the question, its topic and focus, the type of the expected answer are pieces of information extracted by the NL preprocessing phase. Usually, the list of significant words is extended with their synonyms (sometimes with hypernyms as well) and their morphological variants (especially for language with rich morphology). Each such word becomes a search criterion and sometimes it is associated with a numerical score (figure of merit) representing its importance in the meaning of the initial question. After the query is generated in compliance with the syntax and semantics of the query language understood by the search engine, it is further processed by the embedded search engine, which in return delivers a ranked list of all documents assumed to be relevant for the initial question. The search engine may be instructed to return only the N (arbitrary large) top relevant documents or to return only the documents the relevance of which is above a previously established threshold.

This set of documents is further processed by a second NL module in order to detect, extract and rank the sentences or paragraphs that are most likely to provide the answer to the user's initial question (see Figure 1).

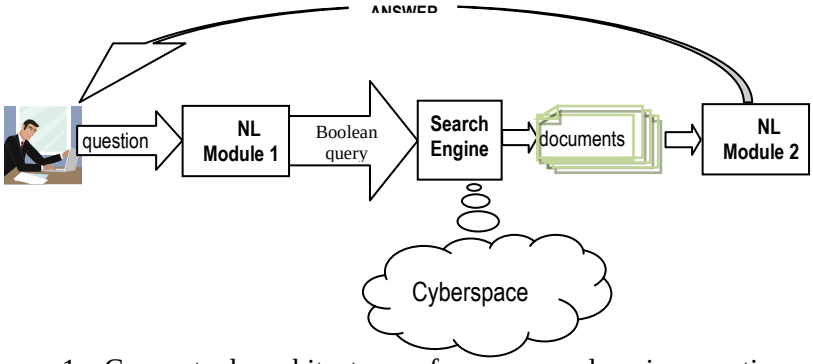

Figure 1: Conceptual architecture of an open domain question-answering system

To be more explicit with respect to the difference between the tasks assumed by an IIR and an ODQAS and to the evaluation of their performances, let us assume a set of N questions $\{Q_1, Q_2, \ldots Q_N\}$ for which the right answers are known, extracted from a large collection of

documents (the entire web, at the limit). Let us further assume that these answers are represented as the set $\left\{D_1 : p_i^{D_1}, D_2 : p_j^{D_2}, ..., D_N : p_k^{D_N}\right\}$, where D_i stands for the document containing the right answer to the question *i* and p_q^{Di} represents the q^{th} textual unit (paragraph, sentence or phrase) in the document D_i that properly answers the question *i*. The set $\{Q_1, Q_2, \ldots Q_N\}$ of questions is given to an IIR system and respectively to a ODQAS and they are expected to return exactly one answer to each question:

IIR system response: $\left\{D_1^{'}, D_2^{'}, ..., D_N^{'}\right\}$

ODQAS response: $\left\{D_1^{''} : p_m^{D_1^{''}}, D_2^{''} : p_n^{D_2^{''}}, ..., D_N^{''} : p_s^{D_N^{''}}\right\}$

One method to evaluate the accuracy of the two sets of answers is described by the equations below:

$$ACC_{IIR} = \frac{1}{N}\sum_{i=1}^{N}\delta(D_i, D_i^{'}),\ ACC_{ODQA} = \frac{1}{N}\sum_{i=1}^{N}\delta(D_i : p_m^{D_i}, D_i^{''} : p_q^{D_i^{''}})$$

where δ (Kronecker's delta) is 1 if its arguments are identical and 0 otherwise. Ideally, ACC_{IIR} and ACC_{ODQAS} should be 1 (that is, the systems should correctly answer all questions). It is easy to see that the evaluation score is tougher for the ODQAS than for the IIR system, because the former has to additionally identify the text unit that answers the question.

Computing the ratio $M = \frac{ACC_{ODQA}}{ACC_{IRR}}$ one could estimate the merits of NLP techniques with respect to the analysed tasks. With a figure of merit M greater or equal than 1 one may say that the NLP methods are undeniably useful, solving better a more difficult problem. This kind of evaluation has been for the first time conducted in Europe at the Cross Language Evaluation Forum (CLEF) in 2009.

2 Evaluation Campaigns in Multilingual Information Retrieval and Natural Language Processing; the Case of Romanian

The evaluation campaigns in the domains of Intelligent Information Retrieval and Natural Language Processing are a constant priority of the advanced research dedicated to the digital knowledge space. They ensure an objective environment for assessing the scientific and technological

advances towards removing the linguistic barriers and the universal access to the knowledge on the web. The first evaluation campaigns were organized in the late 80's in USA (MUC-Message Understanding Conference, TREC-Text Retrieval Conference, DUC-Document Understanding Conference, TAC-Text Analysis Conference), and beginning with year 2000 in Japan (NTCIR) and Europa (CLEF). The languages of interest in these initiatives, besides English, where those for which economic, political or military rationals are high (Japanese, Chinese, Arabic). Later on, some other "big" languages were included into the evaluation campaigns: Spanish, French, German, Russian. With the beginning of the millenium, an interest emerged, also in USA, for (computationally) less-studied languages, generically called "under-resourced" languages. In 2003, at the Conference of the North American Association for Computational Linguistics (NAACL2003, Edmonton, Canada) and in 2005 at the Conference of the Association for Computational Linguistics (ACL2005, Ann Arbor, SUA) there were organized the first evaluation competitions on the task of lexical alignment of bilingual texts, one language being English and the other one being an "under-resourced" language (Hindi, Inuktitut and Romanian).

The official evaluation campaigns on IIR and NLP systems in Europe are organized every year, since 2000, by the European Commission supported Cross Language Evaluation Forum (CLEF). The major focus of CLEF is on the European languages and beginning with 2006 the Member State languages, considered to be under-resourced (Bulgarian, Czech, Estonian, Finnish, Greek, Hungarian, Irish, Latvian, Lithuanian, Maltese, Polish, Portuguese, Romanian, Slovak and Swedish) became eligible for evaluation contest, provided local interest and available running systems existed. Consequently, Romanian language has been enrolled in CLEF competitions every year since 2006. Besides the institutions in Romania involved in NLP research for many years (Research Institute for Artificial Intelligence of the Romanian Academy in Bucharest, University "A.I. Cuza of Iaşi and Institute for Theoretical Informatics of the Romanian Academy in Iasi) a great interest for Romanian language has been shown, by representative institutions from Great Britain (Worverhampton University), Germany (Hamburg University), France (Université du Strasbourg - INSA, Université Marc Bloch, Université du Grenoble), Italy

(ITC IRST Trento, European Commission Joint Research Centre of Ispra), Denmark (Syddansk University), Spain (Alicante University, Cataluña University), SUA (Texas University, California University, Maryland University etc.), Canada (Montreal University) and many others. Automatic processing of Romanian language is an object of investigation also for some of the largest language technology and software companies in the world (Google, Microsoft, Xerox, Language Weaver, etc.).

It is a title of pride in saying that all technical and scientific competitions carried out so far for assessment of NLP systems for Romanian have been won by research teams from Romania: Research Institute for Artificial Intelligence of the Romanian Academy-ICIA/RACAI (NAACL 2003 - Edmonton, Canada; ACL 2005 - Ann Arbor, USA; CLEF 2006 - Alicante, Spain; CLEF 2007- Budapest, Hungary; CLEF 2009 - Kerkyra, Greece) and "A.I. Cuza" University (CLEF 2008 - Aarhus, Denmark, CLEF 2010- Padua, Italy).

Table 1. c©1 for CLEF2009 participating systems according to the language

System	**BG**	**DE**	**EN**	**ES**	**FR**	**IT**	**PT**	**RO**
icia092								**0.68**
nlel092				0.47				
uned092			0.61	0.41				
uned091			0.6	0.41				
icia091								0.58
nlel091			0.58	0,35	0.35	0.52		
uaic092			0.54					0.47
loga091		0.44						
loga092		0.44						
base092	0.38	0.38	0.53	0,4	0.45	0.42	0.49	0.44
base091	0.38	0.35	0.51	0.33	0.39	0.39	0.46	0.37
elix092			0.48					
uaic091			0.44					0.45
elix091			0.42					
mira091				0.32				
mira092				0.29				
iles091					0.28			
syna091			0.28		0.23			

isik091			0.25					
iiit091			0.2					
elix092euen			0.18					
elix091euen			0.16					

The 10 years anniversary CLEF 2009 Edition, organized under the auspices of the European coordination action TrebleCLEF of the European Commission's 7th Framework Programme (FP7), brought a number of methodological innovations, allowing multilingual comparison and evaluation of the processing systems for different languages.

Table 2. M=C@1/Best IR baseline (base092)

System	**DE**	**EN**	**ES**	**FR**	**IT**	**RO**
icia092						**1.55**
icia091						**1.32**
nlel092			1.175			
loga091	1.158					
loga092	1.158					
uned092		1.151	1.025			
uned091		1.132	1.025			
nlel091		1.094	0.875	0.78	1.24	
uaic092		1.019				1.07
elix092		0.906				
uaic091		0.83				1.02
mira091			0.8			
elix091		0.792				
mira092		0.725				
iles091				0.62		
syna091		0.528		0.51		
isik091		0.472				
iiit091		0.377				
elix092euen		0.34				
elix091euen		0.302				

For the first time, the results provided by the systems enrolled in the natural language question-answering competition (CLEF-ResPubliQA) could be cross-lingually compared. The test questions (500) were the same in 8 languages (Basque, Bulgarian, English, French, German,

Italian, Romanian and Spanish) and the answers had to be sought in the parallel corpus of EU law "Acquis Communautaire" available in all EU languages. In addition, the organizers of the competition measured, as described previously, the factors of merit $M = \frac{ACC_{QA}}{ACC_{IRR}}$ for all systems participating in the contest.

They used a "state-of-the-art" intelligent information retrieval system, language independent and without using natural language processing techniques. The responses of this system (base092), computed for all the 8 languages of the competition, were evaluated with the same procedure and used as language specific baselines. The purpose of this initiative was to evaluate, as objectively as possible, the usefulness of advanced NLP techniques in finding the right answers. This benchmarking showed that only 50% of the systems were able to exceed the baseline IIR performances. A detailed analysis is presented in (Peñas et al, 2010).

The system developed at our institute (icia092), described in the rest of this article, got the best scores (c@1 and M) of all systems under evaluation runs: 68% (see Table 9 and 10 in (Peñas et al, 2010) reproduced above as Table 1 and Table 2).

3 The ICIA/RACAI ODQA System[1]

Most of the ODQA systems are based on machine learning techniques which ensure domain independence and scalability. Our system is not an exception and, by appropriate training, it combines in a principled way a set of textual features to derive the relevance scores of the documents, paragraphs and sentences. We were inspired by the Minimum Error Rate Training (MERT) optimization (Och, 2003) where a set of features that are supposed to characterize the translation task are assigned significance weights, the linear combination of which provides a global score. Based on these global scores, the candidates are ranked and the best is provided as a solution to the translation task. In our case, we considered a set of features with relevance to a candidate answer to the user's question. The only impediment in using MERT is that when trying to optimize the response of the QA system on a test set of *N* questions, for each question

[1] This section is based on the description given in (Ion et al., 2010). Meanwhile, the system has been extended with cross-lingual (EN-RO) capabilities and has been trained on more parallel data.

having M candidates that are to be globally scored with m parameters with a 10^{-q} precision, there are $M \cdot N \cdot \binom{m + 10^q - 1}{10^q - 1}$ summations of the type shown by equation 1 below. In this case, in order to determine the value of the parameters and keeping the time complexity in reasonable limits, we implemented a hill climbing algorithm, setting initial values for the parameters with q = 1 and then increase the value of q (with an increment value of 10^{-2}) until the peak of the hill is reached.

Before describing the training procedure and the QA algorithm, we will briefly present the document collection which was used as a search space for the CLEF-ResPubliQA shared task.

3.1 The Document Collection

The document collection was based on a subset of the JRC Acquis corpus (Steinberger et al., 2006) comprising of 10714 pairs of English-Romanian documents conforming to the TEI format specifications[2]. We only took the body of the document into consideration when extracting the text to be indexed. This text has been preprocessed by TTL and LexPar (Tufiş et al., 2008) to obtain POS tagging, lemmatization, chunking and dependency linking.

The body part of one JRC-Acquis document is divided into paragraphs, the unit of text required by the ResPubliQA task to be returned as the answer to the user's question. The specifications of this task define five possible types of questions: "factoid", "definition", "procedure", "reason" and "purpose". The classes "reason" and "purpose" were merged into a port-manteau class "reason-purpose" because we found that our classifier made an unreliable distinction between the two initial classes. By labeling the paragraphs with the type of the expected answer we reduced the complexity of the IR problem: given a query, if its type is correctly identified, the answer is searched through only a portion of the full corpus. We used the maximum entropy method for paragraph classification. For feature selection we differentiated between clue words, morpho-syntactical, punctuation, orthographical and sentence length related features. The classifier was trained on 800 manually labeled paragraphs from the JRC-Acquis and its estimated accuracy is approximately 94%.

[2] http://www.tei-c.org/Guidelines/

The JRC-Acquis documents are manually classified using the EUROVOC thesaurus[3] that has more than 6000 terms hierarchically organized. Considering the fact that the technical terms occurring in the JRC-Acquis were supposed to be translated using the EUROVOC terms, the term list of our tokenizer was extended so that these terms would be later recognized. If a term is identified, it counts as a single lexical token as in "*adunare parlamentară"* ("parliamentary assembly").

3.2 The Workflow of NLP Web Services and the Query generation

The ICIA/RACAI's QA system is practically a workflow built on top of our NLP web services. It is a trainable system that uses a linear combination of relevance features scores s_i of the textual unit p, in order to obtain a global relevance measure $S(p)$ which will be used as the sort key:

$$S(p) = \sum \lambda_i s_i \ , \quad \sum \lambda_i = 1 \qquad (1)$$

where s_i is one of the following feature scores ($s_i \in [0,1]$):

1. an indicator function that is 1 if the estimated class of the question is identical to that of the candidate paragraph or 0 otherwise (let's call this score s_1);
2. a lexical chains based score computed between lemmas of the candidate paragraph and lemmas of the question (s_2);
3. a BLEU-like (Papineni et al., 2002) score that gives more weight to paragraphs that contain keywords from the question in the same order as they appear in the question (s_3);
4. the paragraph and document scores as returned by the search engine[4] (s_4 and s_5).

When the QA system receives an input question, it first calls the TTL web service[5] to obtain POS tagging, lemmatization and chunking. Then, it calls the question classifier[6] to decide on the question class after which two types of queries are computed[7]. Both queries may contain the question

[3] http://europa.eu/eurovoc/

[4] We used the Lucene search engine (http://lucene.apache.org).

[5] http://ws.racai.ro/ttlws.wsdl

[6] http://shadow.racai.ro/JRCACQCWebService/Service.asmx?WSDL

[7] One of the query computation algorithms is also implemented as a web service and it is available at http://shadow.racai.ro/QADWebService/Service.asmx?WSDL

class as a search term to be matched with the class of candidate paragraphs. The search engine[8] will return two lists L_1 and L_2 of at most 50 paragraphs that will be sorted according to the eq. 1. The answer is a paragraph p from both L_1 and L_2 for which

$$\text{argmin}_p[\text{rank}_1(p) + \text{rank}_2(p)], \qquad \text{rank}_{1,2}(p) \leq K,\ K \leq 50 \qquad (2)$$

where $\text{rank}_j(p)$ is the rank of paragraph p in L_j. Experimenting with different values for *K* on an in-house developed 200 questions test set (see below), we determined that the best value for *K* is 3. When such a common paragraph does not exist, the system returns the *no answer* (NOA[9]) string.

Our QA system is trainable in the sense that the weights (λ_i) that we use to combine our relevance features scores are obtained through a MERT-like optimization technique. For the training the ranking parameters we used the *Mean reciprocal rank* (MRR), a statistic for evaluating any process that produces a list of possible responses to a query, ordered by probability of correctness. The reciprocal rank of a query response is the multiplicative inverse of the rank of the first correct answer. The mean reciprocal rank is the average of the reciprocal ranks of results for a sample of queries Q (Vorhees, 1999): $\frac{1}{|Q|}\sum_{i-1}^{Q}\frac{1}{rank_i}$.

Since the development question set comprised of only 20 questions, we proceeded to the enlargement of this test set (having the 20 questions as examples). We produced another 180 questions to obtain a new development set of 200 questions simply by randomly selecting documents from the JRC-Acquis corpus and reading them. For each question we provided the ID of the paragraph that contained the right answer and the question class. The training procedure consisted of:

- running the QA system on these 200 questions and retaining the first 50 paragraphs for each question according to the paragraph score given by the search engine (s_4);
- obtaining for each paragraph the set of 5 relevance scores, $s_1 \ldots s_5$;

[8] http://www.racai.ro/webservices/search.asmx?WSDL

[9] The ResPubliQA organizers considered that instead of a wrong answer a No Answer (NOA) is more useful, not only because the user is not misguided but also because such an answer underlines a kind of self-aware of the QA system on its answering accuracy.

for each combination of λ parameters with $\sum_{i=1}^{5} \lambda_i = 1$ and increment step of 10^{-2}, compute the Mean Reciprocal Rank (MRR) of the 200 question test set by sorting the list of returned paragraphs for each question according to eq. 1;

- retaining the set of λ parameters for which we obtain the maximum MRR value.

The two QA systems (each one corresponding to specific algorithm of query generation) were individually optimized with no regard to NOA strings and we added the combination function (eq. 2) in order to estimate the confidence in the chosen answer (an optional requirement of the ResPubliQA task).

The first algorithm of query generation (the TFIDF query algorithm) considers all the content words of the question (nouns, verbs, adjectives and adverbs) out of which it constructs a disjunction of terms (which are lemmas of the content words) with the condition that the TFIDF of the given term ***t*** is above a certain threshold:

$$\mathrm{TFIDF}(t) = (1 + \ln(f_{\mathrm{t}})) \cdot \ln\left(\frac{D}{f_{\mathrm{d}}}\right) \qquad (3)$$

in which 'ln' is the natural logarithm, f_{t} is the term frequency in the entire corpus, f_{d} is the number of documents in which the term appears and D is the number of documents in our corpus, namely 10714 (if f_{t} is 0, f_{d} is also 0 and the whole measure is 0 by definition). The rationale behind this decision is that there are certain terms that are very frequent and also very uninformative.

The second algorithm of query generation (the chunk-based query algorithm) also uses the TTL preprocessing of the question. The algorithm takes into account the noun phrase (NP) chunks and the main verbs of the question. For each NP chunk, two (instead of one) query terms are constructed: (i) one term is a query expression obtained by concatenating the lemmas of the words in the chunk and having a boost equal to the number of those words, and (ii) the other one is a Boolean query in which all the different lemmas of the words in the chunk are joined by the conjunction operator. For example an "a b c" chunk generates the following two queries: "l(a) l(b) l(c)"^3 and (l(a) AND l(b) AND l(c)) where l(w) is the lemma for the w word. For each chunk of length n, we generate all the sub-chunks of length $n-1, n \geq 2$ (i.e. "a b" and "b c") and apply the same steps.

As already stated, the QA system uses a linear combination of relevance features scores (eq. 1) to score a given candidate paragraph as to the possibility of containing the answer to the question. The BLUE-like similarity measure (s_3) between the question and one candidate paragraph stems from the fact that there are questions that are formulated using a high percentage of words in the order that they appear in the answer containing paragraph. BLEU is a measure that counts *n*-grams from one candidate translation in one or more reference translations. We use the same principle and count *n*-grams from the question in the candidate paragraph but here is where the similarity to BLEU ends. Our *n*-gram processing counts only content word *n*-grams (content words are not necessarily adjacent). Actually, an *n*-gram is a sliding window of question content word lemmas of a maximum length equal to the length of the question (measured in content words) and a minimum length of 2.

4 Evaluations

Each query produces a different set of paragraphs when posed to the search engine thus allowing us to speak of two different QA systems. We applied the training procedure described in the previous section on our 200 questions test set with each system and ended up with the following values for the λ parameters:

Table 3. Parameters for paragraph score weighting

	λ_1	λ_2	λ_3	λ_4	λ_5
The TFIDF query algorithm	0.22	0.1	0.1	0.19	0.39
The chunk query algorithm	0.32	0.14	0.17	0.25	0.12

With these parameters, each system was given the official ResPubliQA 500 questions test set. For each question, each system returned 50 paragraphs that were sorted according to eq. 1 using parameters from Table 3. Table 1 contains the official evaluations (Peñas et al., 2010) of our two runs, ICIA091RORO and ICIA092RORO. The first run, officially rated with the fourth c@1 score, corresponds to running the two QA systems with queries exactly as described. The second run, officially rated with the best c@1 score, was based on queries that included the class of the question as a search term. When we constructed the index of paragraphs we added a field that kept the

paragraph class. This additional search term brought about a significant improvement in both accuracy and c@1 measure as Table 1 shows.

A very interesting evaluation performed by the organizers was to estimate the accuracy improvement of the NLP QA systems as compared to language-specific baseline IR systems. According to this new evaluation both ICIA runs received the highest scores out of the evaluated runs (see Table 2).

After the official competition was closed and the evaluation results published, we continued to make various experiments with respect to the optimal values of the λ parameters. The values in Table 3 did not take into account the question class. We hypothesized that training different sets of λ parameters for each QA system and for each question class would yield improved results. We experimented with our 200 questions test set and trained different sets of parameters (with the increment step of 0.05 to reduce the time complexity) for each question class and our expectations were met. Both QA (icia092 and icia091) increased their c@1 scores with 2.3% and 1.8% respectively. Table 4 presents the optimally trained values for the five λ parameters when taking into account the question types.

Table 4. Different parameters trained for different classes

		λ_1	λ_2	λ_3	λ_4	λ_5
The TFIDF query algorithm	Factoid	0.1	0	0.2	0.4	0.3
	Definition	0.2	0.15	0.05	0.15	0.45
	Reason	0.1	0	0.15	0.3	0.45
	Procedure	0.1	0	0.15	0.15	0.6
The chunk query algorithm	Factoid	0.15	0	0.3	0.3	0.25
	Definition	0.05	0.5	0.15	0.1	0.2
	Reason	0.2	0	0.4	0.2	0.2
	Procedure	0.15	0.1	0.25	0.2	0.3

5 Conclusion

The CLEF campaign has already a good tradition in evaluating NLP and IIR systems. Each year, the evaluation exercise showed its participants how to test and then, how to improve their systems. The competitive framework has motivated systems designers to adopt daring solutions and to experiment in order to obtain the best result. The evaluation of only the first answers (due to limited resources) underestimates the utility of the

ODQA systems because the interested user will be always ready to inspect, say, 5 top returned results. The probability of having the correct answer in one of the five top results is significantly higher, in our case coming close to 92%.

The QA system presented in this article is evolving into a cross-lingual question answering system capable to receive questions in one language, currently Romanian or English, and look for the answers in documents written in one of the two languages, irrespective of the interrogation language. We have already processed the English side of the JRC-Acquis and, given that we have several functional Example-Based and Statistical Machine Translation Systems (Irimia, Ceauşu,2010) we plan to incorporate into the workflow a translation module either for the natural language question or for the generated query. Then the combination method expressed by eq. 2 is expected to yield better results if applied on English and Romanian paragraph lists since a common paragraph means the same information found via two different languages. This estimation is strengthened by the analysis made by the ResPubliQA organizers (Peñas, et al., 2010), according to which 99% of questions have been correctly answered by at least one system in at least one language.

The principal advantage of this approach to QA is that one has an easily extensible and trainable QA system. If a new way to assess the relevance of a paragraph to a given question comes out, then we simply add another parameter that will account for the importance of that measure and retrain.

Acknowledgments. The work reported here was funded by the SIR-RESDEC project, under the grant no. 11-007, and the most recent developments are supported by the STAR project, under the grant no. ID. 1443, both financed by the Ministry of Education, Research and Innovation.

References

[1] Ion, R., Ştefănescu, D., Ceauşu, Al., Tufiş, D., Irimia, E., Barbu Mititelu, V. (2010): *"A Trainable Multi-factored QA System"* In Carol Peters et al. (eds.) Multilingual Information Access Evaluation, Vol. I Text Retrieval Experiments, pp. 257-264, Lecture Notes in Computer Science, Volume 6241, Springer-Verlag. ISBN: 978-3-540-85759-4.

[2] Irimia, E., and Ceaușu, Al. (2010). *Dependency-based translation equivalents for factored machine translation*, In Alexander Gelbukh (ed.) Research In Computer Science, Special Issue on NLP and its Applications 46, pp. 205—216, ISSN: 1870-4069.

[3] Och, F.J. (2003): *Minimum Error Rate Training in statistical machine translation.* In: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics, pp.160—167. Sapporo, Japan, July 07-12.

[4] Papineni, K., Roukos, S., Ward, T., Zhu, W.J. (2002): *BLEU: a method for automatic evaluation of machine translation.* In: Proceedings of the ACL-2002, 40th Annual meeting of the Association for Computational Linguistics, pp. 311–318. Philadelphia, July.

[5] Peñas, A., Forner, P., Sutcliffe, R., Rodrigo, Á., Forăscu, C., Alegria, I., Giampiccolo, D., Moreau, N., Osenova, P: *Overview of ResPubliQA 2009 (2010): Question Answering Evaluation over European Legislation.* In Carol Peters et al. (eds.) *Multilingual Information Access Evaluation, Vol. I Text Retrieval Experiments*, pp. 174-196, Lecture Notes in Computer Science, Volume 6241, Springer-Verlag. ISBN: 978-3-540-85759-4.

[6] Steinberger, R., Pouliquen, B., Widiger, A., Ignat, C., Erjavec, T., Tufiș, D., Varga, D. (2006). *The JRC-Acquis: A Multilingual Aligned Parallel Corpus with 20+ Languages.* In Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006), pp. 2142-2147, Genoa, Italy, May 2006. ELRA - European Language Resources Association.

[7] Tiedemann, J. (2009). *News from OPUS - A Collection of Multilingual Parallel Corpora with Tools and Interfaces.* In N. Nicolov and K. Bontcheva and G. Angelova and R. Mitkov (eds.) Recent Advances in Natural Language Processing (vol V), pages 237-248, John Benjamins, Amsterdam/Philadelphia.

[8] Tufiș, D., Ion, R., Ceaușu, Al., and Ștefănescu, D. (2008). *RACAI's Linguistic Web Services.* In Proceedings of the 6th Language Resources and Evaluation Conference - LREC 2008, Marrakech, Morocco, May. ELRA - European Language Resources Association. ISBN 2-9517408-4-0.

[9] Voorhees, E., M. (1999). "Proceedings of the 8th Text Retrieval Conference". TREC-8 Question Answering Track Report. pp. 77–82.

Dan Tufiș[1]

[1] Research Institute for Artificial Intelligence, Romanian Academy 13, Calea 13 Septembrie, Bucharest 050711, Romania

E-mail: tufis@racai.ro