JU_CSE_TE: System Description QA@CLEF 2010 – ResPubliQA

Partha Pakray[1], Pinaki Bhaskar[1], Santanu Pal[1], Dipankar Das[1],
Sivaji Bandyopadhyay[1], Alexander Gelbukh[2]

Department of Computer Science & Engineering[1],
Jadavpur University, India[1]
Center for Computing Research[2],
National Polytechnic Institute, Mexico City, Mexico[2]
{parthapakray, pinaki.bhaskar, santanu.pal.ju,
dipankar.dipnil2005@gmail.com}@gmail.com[1], sivaji_cse_ju@yahoo.com[1],
gelbukh@gelbukh.com[2]

Abstract. The article presents the experiments carried out as part of the participation in the Paragraph Selection (PS) Task and Answer Selection (AS) Task of QA@CLEF 2010 – ResPubliQA. Our System use Apache Lucene for document retrieval system. All test documents are indexed using Apache Lucene. Stop words are removed from each question and query words are identified to retrieve the most relevant documents using Lucene. Relevant paragraphs are selected from the retrieved documents based on the TF-IDF of the matching query words along with n-gram overlap of the paragraph with the original question. Chunk boundaries are detected in the original question and key chunks are identified. Chunk boundaries are also detected in each sentence in a paragraph. The key chunks are matched in each sentence in a paragraph and relevant sentences are identified based on the key chunk matching score. Each question is analyzed to identify its possible answer type. The SRL Tool (Assert Tool Kit) [1] is applied on each sentence in a paragraph to assign semantic roles to each chunk. The Answer Extraction module identifies the appropriate chunk in a sentence as the exact answer whose semantic role matches with the possible answer type for the question. The tasks have been carried out for English. The Paragraph Selection task has been evaluated on the test data with an overall accuracy score of 0.37 and c@1 measure of 0.50. The Answer Extraction task has performed poorly with an overall accuracy score of 0.16 and c@1 measure of 0.26.

Keywords: Lucene Index, Chunk Boundary, n-gram overlapping, SRL Tool

1 Introduction

After the success of ResPubliQA 2009 [2], the organizers announced the ResPubliQA 2010, the second evaluation campaign of Question Answering system over European Legislation to hold it within the framework of CLEF 2010 conference. The main goal

of ResPubliQA 2010[1] is to find the appropriate single paragraph that contains the answer along with the exact answer of a given question from a collection of parallel documents in the European languages.

The aim of ResPubliQA 2010 [3] is to capitalize on what has been achieved in the previous evaluation campaign while at the same time adding a number of refinements:

- The addition of new question types and the refinement of old ones;
- The opportunity to return both paragraph and exact answer;
- The addition of a new document collection: EUROPARL.

Two separate tasks are part of the ResPubliQA 2010 [3] evaluation campaign:

i. PARAGRAPH SELECTION (PS) TASK: to retrieve one paragraph (Text+ID) containing the answer to a question in natural language. This task is very similar to the one performed in 2009.

ii. ANSWER SELECTION (AS) TASK: beyond retrieving a paragraph, systems are required to retrieve also the exact answer (shorter string of text) answering a question in natural language.

The parallel-aligned documents are available in 9 languages, i.e. Bulgarian, Dutch, English, French, German, Italian, Portuguese, Romanian and Spanish.

2 Corpus Statistics

The ResPubliQA [3] collection is made up of a subset of two multilingual parallel aligned document collections.

i. **The JRC-ACQUIS Multilingual Parallel Corpus**[2]: The JRC-ACQUIS corpus contains the complete EU legislation, including texts between the years 1950 to 2006. A sub-set of the JRC-ACQUIS has been created with roughly 10,700 parallel and aligned documents in each of the 9 languages involved in the track.

ii. **The Europarl collection**[3]: A (very small) subset of the Europarl corpus has been created with parallel documents in all the 9 languages involved in the track by crawling the web to get the data from the European Parliament's website. The sub-set includes 150 parallel and aligned documents per language.

The subject of the JRC-ACQUIS documents is European legislation while the EUROPARL collection deals with the parliamentary domain.

3 System Framework

In this section, we describe our Information Retrieval (IR) based Question Answering (QA) system. The system is defined in three parts: documents selection from indexed documents in the collections, paragraph selection from documents and finally answer selection from the paragraph.

1 http://celct.isti.cnr.it/ResPubliQA/index.php
2 http://wt.jrc.it/lt/Acquis/
3 http://www.europarl.europa.eu/

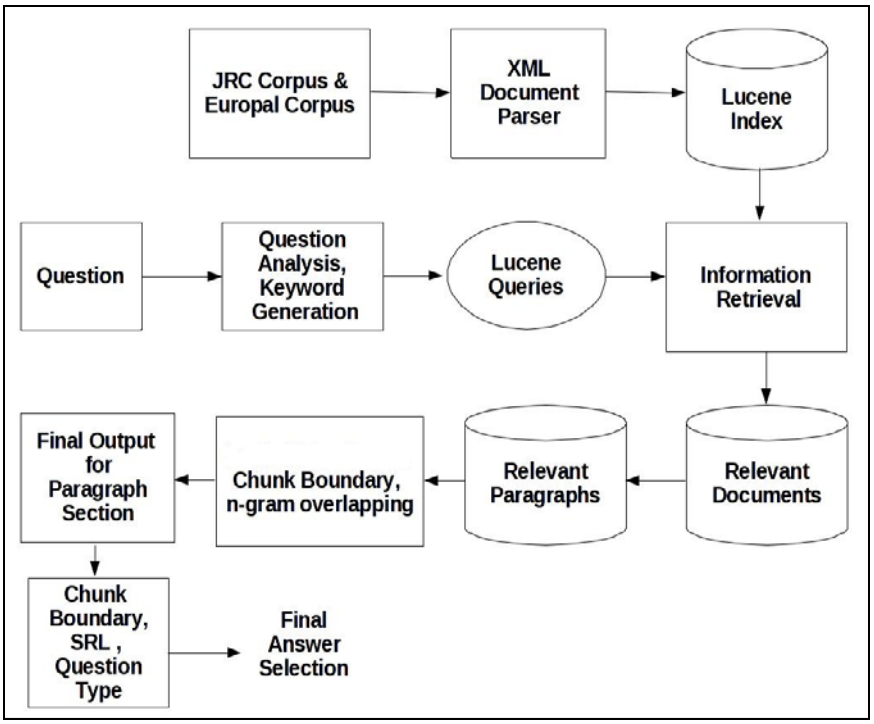

Fig. 1. IR based QA Model

The Apache Lucene[4] IR system has been used for the present task. Lucene follows the standard IR model with Document parsing, Document Indexing, TF-IDF calculation, query parsing and finally searching/document retrieval. Some modules in Lucene have been upgraded for our present need as described below.

3.1 Document Parsing

The web documents are full of noises mixed with the original content. In that case it is very difficult to identify and separate the noises from the actual content. ResPubliQA 2010 Corpus had many noise in the documents and the documents are in tagged format. So, first of all the documents had to be preprocessed. The document structure is checked and reformatted according to the system requirements.

4 http://lucene.apache.org/

3.1.1 XML Parser. The corpus was in XML format. All the XML test data has been parsed before indexing using our XML Parser. The XML Parser extracts the Title of the document along with the paragraphs.

3.1.2 Remove Noise and Symbols. The corpus has some noise as well as some special symbols that are not necessary for our system. The list of noise symbols and the special symbols is initially developed manually by looking at a number of documents and then the list is used to automatically remove such symbols from the documents. Table 1 lists some of the noisy tokens and their replacements.

Table 1. Token Replacement List

Replace by blank	Replace by Symbol	
	Original Token	Replaced Token
. –	á	a
();	č	c
[...]	è	e
()	š	s

3.2 Document Indexing

After parsing the documents, they are indexed using Lucene, an open source full text search tool.

3.3 Question Processing for query word identification

After indexing has been done, the queries have to be processed to retrieve relevant documents. Each question is processed to identify the query words for submission to Lucene. The question processing steps are described below:

3.3.1 Key-character Removal. Certain key characters in the query cause implicit query handling during searching like dot character between two query words denotes AND of the two query words. Such key characters are thus removed from the question before submission to Lucene. For example, http://wt.jrc.it/ = “http wt jrc it”, doug@nutch.org = “doug nutch org”, etc.

3.3.2 Stop Word Removal. In this step the query words are identified from the question. The Stop words and question words (what, when, where, which etc.) are removed from each question and the words remaining in the question after the

removal of such words are identified as the query words. The stop word list used in the present work can be found at http://members.unine.ch/jacques.savoy/clef/.

3.3.3 Stemming. Query words may appear in inflected forms in the question. For English, standard Porter Stemming algorithm [5] has been used to stem the query words.

3.4 Document Retrieval

After searching each query into the Lucene index, a set of retrieved documents in ranked order for each query is received.

First of all, all queries were fired with AND operator. If at least one document is retrieved using the query with AND operator then the query is removed from the query list and need not be searched again. The rest of the queries are fired again with OR operator. OR searching retrieves at least one document for each query. Now, the top ranked relevant ten documents for each query are considered for Paragraph selection. In case of AND search only the top ranked document is considered. Document retrieval is the most crucial part of this system. We take only the top ranked relevant documents assuming that these are the most relevant documents for the query or the question from which the query had been generated.

3.5 Relevant Paragraph Selection

The selection of relevant paragraphs is one of the important activities of this system. We have used both "AND" and "OR" searching similar to document retrieval, to select relevant paragraphs from each retrieved relevant document. First those paragraph(s) are identified that contain all the query words. If at least one paragraph containing all the query words is found then the paragraph selection process for that document is stopped. Otherwise, we continue searching the paragraphs which contain at least one query word. Such relevant paragraphs are ranked using the n-gram overlap score between the paragraph and the original question. By the above process all the relevant paragraphs for each query are identified.

3.5.1 n-gram Overlap. In this step the n-grams are identified from the question. These n-grams from the question are matched in the documents. If no match is found for a higher order n-gram then the search is repeated for the immediate lower order n-gram. For each n-gram overlap, the score is calculated as the value of n plus n/100. The additional score of n/100 assures that higher order n-gram overlap will have a higher bonus in the score. The composite n-gram overlap score for a paragraph is the sum of the individual n-gram overlap scores. The paragraph that has the highest n-gram overlap score is selected as the answer paragraph. If more than one paragraph

[5] http://tartarus.org/~martin/PorterStemmer/java.txt

has the same highest score then the paragraph that occurs earlier in the document is selected as the answer paragraph.

3.6 Question Analysis

The question sentences are pre-processed using Stanford Dependency parser [4]. The words along with their part of speech (POS) information are passed through a Conditional Random Field (CRF) based chunker [5] to extract phrase level chunks of the questions. A rule-based module is developed to identify the chunk boundaries. Key chunks are identified for each question. The chunks that are related by each *prep* relation constitute the key chunks corresponding to that *prep* relation. These key chunks are searched in the answer paragraph. We analyze each question to identify its possible answer type based on the question keyword as listed in Table 2.

Table 2. Question Keyword and Expected Answer

Question Type	Expected Answer
Who	PERSON
When	DATE / TIME
Where	LOCATION
Why	REASON
What	OBJECT / DEFINITION
How	MEASURE

3.7 Answer Sentence Selection in a Paragraph

The sentences in the answer paragraph are detected. Each sentence is processed using Stanford Dependency parser and chunker as well. Our chunk boundary detector module detects every chunk as well as its boundary in each and every sentence. Each sentence is assigned a score based on the matching of question key chunks in the sentence. The top ranked sentence in each answer paragraph is identified as the answer sentence.

Each such answer sentence in the answer paragraph is passed to the SRL Tool Kit [1] for appropriate labeling of the semantic roles to each chunk in the sentence. The semantic roles ARGM-TMP and ARGM-LOC associated to the chunks help to identify the DATE and LOCATION named entities.

3.8 Answer Selection from paragraph

If the question type is “who”, the answer sentence is passed to the RASP parser [6] mainly to identify the occurrence of PERSON type named entities in the sentence. The PERSON type named entity is identified as the answer phrase. Answers to “when” type of questions are selected by looking for a chunk in the answer sentence that has been labeled with the semantic role ARGM-TMP. In case of “where type”

questions, the chunk in the answer sentence that has been labeled with semantic role ARGM-LOC is identified as the answer phrase. Answers to “what” type questions are identified by looking for cue phrases such as “defined as”, “means that” etc. and then selecting the part of the sentence after the cue phrase till the end of the sentence. In case of ‘why” type questions, the answers are identified by looking for cue phrases like “reason of”, “because of” etc. and then selecting the part of the sentence after the cue phrase till the end of the sentence.

In case of “How much” or “How many” question types, clause detection in the answer sentence becomes necessary as most often these sentences are complex in structure. The punctuation marks, discourse markers identified through mark type dependency relations, causal words (as, because) are used for clause detection. The dependency relations connected directly with each verb chunk are used in clause detection. Each verb chunk and the associated chunks whose head is directly linked with the verb chunk in any dependency relation identify a clause. If any chunk contains any word with POS category CD, the chunk is considered as the answer phrase for the specific question. Otherwise, the candidate phrases that contain capitalized words or Named Entities are considered as the answer to the question. Two examples of Answer Extraction are given below. Only the answer sentence has been shown in these examples and not the answer paragraph.

Table 3. Example of answer extraction

Example: 1
Question (Qid: 0025): When did Dow Chemical obtain the shares of Union Carbide? <p_id="14"> (3) However, since the Dow Chemical Company acquired **on 6 February 2001** all shares of Union Carbide Corporation, a company benefiting from an individual anti-dumping duty of EUR 59,25 per tonne, the Dow Chemical Company is still active in the ethanolamine business.   Question Type: When Expected Answer type: DATE Parse: SRL Tool Answer: **on 6 February 2001**
Example: 2
Question (qid: 0020): How many transactions can be covered in a DEPBS credit application? Chunked Sentence from Parsed output: (How/WRB/B-NP#many/JJ/I-NP#transactions/NNS/I-NP) (can/MD/B-VP#be/VB/I-VP#covered/VBN/I-VP) (in/IN/B-PP) (a/DT/B-NP#DEPBS/NNP/I-NP#credit/NN/I-NP#application/NN/I-NP) (?/./B-O#)   (55) An **application** for **DEPBS credits** can **cover** up to **25** export **transactions** and, if electronically filed, an unlimited amount of export transactions.

Capitalized Phrase: **DEPBS**
Named Entity: **DEPBS**
Verb: **cover**
POS Tag (CD): **25**
Cue Phrase: **DEPBS credits**

Answer: An application for DEPBS credits can cover up to 25 export transactions.

5 Evaluation

We submitted English monolingual run for one Paragraph selection Task and one Answer selection Task. The main measure used in this evaluation campaign is c@1 which is defined in equation 1.

$$c@1 = \frac{1}{n}(n_R + n_U \frac{n_R}{n}) \quad (1)$$

where, n_R: the number of correctly answered questions, n_U: number of unanswered questions and n: the total number of questions

In addition to computing the c@1 score, the answer extraction performance has also been measured as shown in equation 2.

Answer extraction performance= #R / (#R + #X + #M) (2)

where, #R, #X and #M denote the number of answered questions identified as Right, inexact and Missed respectively.

- **Accuracy Measure of Paragraph Selection Task (PS)**:

Our PS file contains a total of 200 answers.
- Number of questions ANSWERED: 125
- Number of questions UNANSWERED: 75
- Number of questions ANSWERED with RIGHT candidate answer: 73
- Number of questions ANSWERED with WRONG candidate answer: 52
- Number of questions UNANSWERED with RIGHT candidate answer: 0
- Number of questions UNANSWERED with WRONG candidate answer: 0
- Number of questions UNANSWERED with EMPTY candidate: 75

The statistics of Paragraph Selection Task (PS) task is given in figure 2.

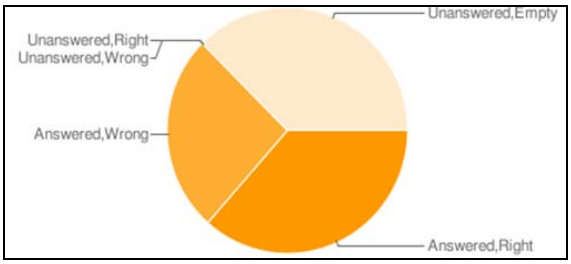

Fig. 2. Statistics of Paragraph Selection Task (PS)

The accuracy of the answer selection process has been calculated as:
Overall accuracy = 73/200 = 0.37
Proportion of answers correctly discarded: 0/75 = 0.00
c@1 measure = (73+75(73/200))/200 = 0.50

- **Accuracy Measure of Answer Selection Task (AS):**

Our AS file contains a total of 200 answers.
- Number of questions ANSWERED: 43
- Number of questions UNANSWERED: 115
- Number of questions ANSWERED with RIGHT candidate answer: 31
- Number of questions ANSWERED with WRONG candidate answer: 12
- Number of questions ANSWERED with MISSED candidate answer: 10
- Number of questions ANSWERED with INEXACT candidate answer: 8
- Number of questions UNANSWERED with RIGHT candidate answer: 0
- Number of questions UNANSWERED with WRONG candidate answer: 40
- Number of questions UNANSWERED with MISSED candidate answer: 24
- Number of questions UNANSWERED with INEXACT candidate answer: 0
- Number of questions UNANSWERED with EMPTY candidate: 75

The statistics of Answer Selection Task (AS) task is given in figure 3.

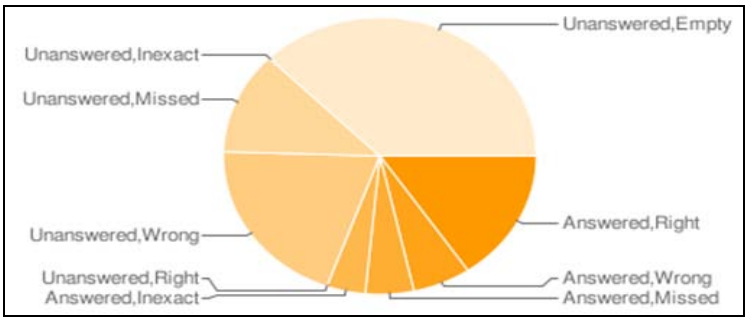

Fig. 3. Answer Selection Task (AS)

Accuracy (unanswered,judgment=correct + answered,judgment=correct) calculated over all assessed answers:

Overall accuracy = 31/200 = 0.16

Proportion of answers correctly discarded: 40/115 = 0.35

c@1 measure = (31+139(31/200))/200 = 0.26

Answer extraction performance = (31/(31+8+10) = 0.63

5 Conclusion

The question answering system has been developed as part of the participation in the ResPubliQA 2010 track as part of the CLEF 2010 evaluation campaign. The system uses document retrieval using Lucene search engine, an n-gram based match for paragraph selection and combines various NLP tools for answer selection. The overall system has been evaluated using the evaluation metrics provided as part of the ResPubliQA 2010 track. The evaluation results are satisfactory considering that this is the first participation in the track. Future works will be motivated towards improving the performance of the system.

Acknowledgement

The work has been carried out with support from IFCPAR funded Project "An Advanced platform for question answering systems" (Project No. 4200-IT-1).

References

1. Sameer S. Pradhan, Wayne Ward, Kadri Hacioglu, James H. Martin, Daniel Jurafsky.: Shallow Semantic Parsing using Support Vector Machines. In Proceedings of the Human Language Technology Conference/North American chapter of the Association for Computational Linguistics annual meeting (HLT/NAACL-2004),Bosto, MA, May 2-7, 2004
2. Anselmo Peñas, Pamela Forner, Richard Sutcliffe, Álvaro Rodrigo, Corina Forăscu, Iñaki Alegria, Danilo Giampiccolo, Nicolas Moreau, Petya Osenova.: Overview of ResPubliQA 2009: Question Answering Evaluation over European Legislation. In Working Notes for the CLEF 2009 Workshop, 30 September-2 October, 2009, Corfu, Greece.
3. Anselmo Peñas, Pamela Forner, Álvaro Rodrigo, Richard Sutcliffe, Corina Forăscu and Cristina Mota.: Overview of ResPubliQA 2010: Question Answering Evaluation over European Legislation. In Working Notes for the CLEF 2010 Workshop, Padua, Italy, 20-23 September 2010.
4. Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning.: Generating Typed Dependency Parses from Phrase Structure Parses. In 5th International Conference on Language Resources and Evaluation (LREC) (2006)
5. Xuan-Hieu Phan.: CRFChunker: CRF English Phrase Chunker. PACLIC 2006. (2006)
6. E. Briscoe, J. Carroll, and R. Watson.: The Second Release of the RASP System. In Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions.