A Question Answering System based on Information Retrieval and Validation

Álvaro Rodrigo, Joaquín Pérez-Iglesias,
Anselmo Peñas, Guillermo Garrido, Lourdes Araujo

NLP & IR Group, UNED, Madrid
{alvarory,joaquin.perez,anselmo,ggarrido,lurdes}@lsi.uned.es

Abstract. Our participation at ResPubliQA 2010 was based on applying an Information Retrieval (IR) engine of high performance and a validation step for removing incorrect answers. The IR engine received additional information from the analysis of questions, what produces a slight improvement in results. However, the validation module discarded sometimes too much correct answers, contributing to reduce the overall performance. These errors were due to the application of too strict constraints. Therefore, future work must be focused on reducing the amount of false negatives returned by the validation module. On the other hand, we observed that IR ranking offers important information for selecting the final answer, but better results could be obtained if additional sources of information were also considered.

1 Introduction

The NLP & IR group at UNED participated at ResPubliQA 2010 after the successful results of its previous participation. The system used in 2009 was based on an Information Retrieval step of high performance and Answer Validation.

ResPubliQA 2010 proposed two tasks related to Question Answering (QA): one for retrieving a paragraph with a correct answer given a question, and a second one where both the paragraph and the exact answer string must be returned. Both tasks were developed using the same set of questions and over the same collections (JRC Acquis[1] and EuroParl[2]). We have participated in monolingual English and Spanish Paragraph Selection (PS) tasks.

This year we proposed to improve the Information Retrieval (IR) step by adding information about the question. Thus, we wanted to increase the recall of the IR engine as well as increase the ranking given to promising candidate paragraphs. Furthermore, we applied an Answer Validation (AV) similar to the one performed last year, including some minor changes for solving some errors. This validation was focused on removing paragraphs that show evidences of not having a correct answer.

The structure of this paper is as follows: the main components of our system are described in Section 2. The description of the runs sent to ResPubliQA is given in Section 3, while the results of these runs are shown in Section 4 and their analysis in Section 5. Finally, some conclusions and future work are given in Section 6.

[1] http://wt.jrc.it/lt/Acquis/

[2] http://www.europarl.europa.eu/

2 System Overview

This section describes the main components of our QA system. Figure 1 shows the architecture of the system. The different phases of the system work for guiding the search to the most promising answers, removing the ones that are considered incorrect.

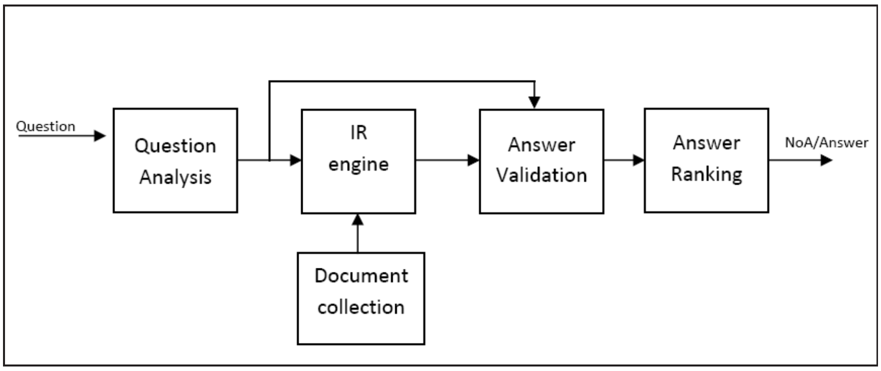

Fig. 1. Architecture of the system

The following subsections describe in detail each one of the different components of the system.

2.1 Question Analysis

The objective of this step is to obtain features from the question that could be helpful in the following steps. All the information obtained by this module is given to the following steps of the system.

The information extracted is:

- The expected answer type, which is an information that offers an important constraint to be accomplish by correct answers. We performed a classification based on handmade patterns where the categories were: *count*, *time*, *location*, *organization*, *person*, *definition* and *other*.
- The question focus, which is a word close to the interrogative term that supplies additional information about the type of the expected answer. The detection of the focus is important for extracting the answer from candidate paragraphs. However, as we participated this year only at the PS task, we used the question focus for other purposes.
  The question focus defines sometimes the context of the question and it is likely that the focus does not appear close to the correct answer. For example, if we have the question *What country was Nadal born in?*, the question focus is *country* and it is likely that a correct answer to this question does not contain the word *country*. Therefore, we used this intuition for creating the query which will be submitted to the IR engine (more details are given in Section 2.2).

- The Named Entities (NE) contained in the question. These NEs are important for supporting the correctness of an answer contained in a candidate paragraph. Hence, NEs represent an important information for detecting correct answers.

2.2 Information Retrieval

The mission of the IR module is to perform a first selection of paragraphs that are considered relevant to the input question. We decided to use BM25 [3] last year, a model that can be adapted to fit the specific characteristics of the data in use. More information about the implementation and successful results of last year IR engine are given in [2].

We decided to keep this model adding some minor modifications with the purpose of improving the recall and ranking of correct paragraphs. The modifications added this year were related to the creation of the query submitted to the IR engine from the input question. These changes were:

- As it was mentioned in Section 2.1, the question focus usually does not appear in correct answers. Thus, we consider that the presence of the focus has to receive a lower importance in the query.
- The NEs of a question represent important information and it is likely they appear in correct answers. Therefore, our intuition was to give a higher importance to NEs in the query.

The procedure for including these two new features was to assign different boost factors to the terms of a query. Then, given a question we built the corresponding query to be used in the IR phase following these steps:

1. Removal of stopwords
2. Stemming pre-process based on Snowball implementation of Porter algorithm
3. Use of different weights, considering three possibilities:
   - high discriminative power: this value is given to NEs
   - medium discriminative power: this value is given to the rest of the terms of the query
   - low discriminative power: this value is given to the question focus (if it exists)

In order to select the values for the different boost factors, we performed several experiments at the development period. We selected the following values after performing several experiments:

- High discriminative terms received boost factor 2.
- Low and normal discriminative terms received boost factor 1. We decided to give the same boost factor to these terms because a lower boost factor of the focus produced worse results in the development period.

The IR engine returned a maximum of 100 candidate paragraphs to the following steps of the system.

2.3 Answer Validation

The mission of this step is to eliminate possible incorrect paragraphs contained in the list returned by the IR engine. Thus, there are more possibilities of giving at the end of the process a correct answer. We say that this module validates a paragraph when it is considered that the paragraph is correct. If a paragraph is considered as incorrect, we say that the paragraph is rejected.

This phase works in a pipeline processing, where a set of constraints are checked for each candidate paragraph in each step. Only candidate paragraphs that accomplish all the constraints are returned at the end of this pipeline.

It is important to remark that this phase is not focused on checking the correctness of a candidate paragraph. It is focused on detecting paragraphs which show some feature that leads to think that they are incorrect. The module was implemented in this way because that it is usually easier to detect incorrect answers than to detect correct ones.

We applied the same three modules used in the last edition. Next sections describe each of these modules in short. More details can be seen in [5].

Expected Answer Type Only paragraphs that contain a NE of the same type that the expected answer type are validated. This validation is performed only for questions where the expected answer was *count*, *time*, *location*, *organization* or *person*. All the paragraphs given to other types of questions are validated by this module.

The Named Entity Recognizer (NER) gave us the distinction among *location*, *organization* and *person* entities only in Spanish. This is why we performed two kinds of matching:

- Fine grained matching: *location*, *organization* and *person* questions must be answered by paragraphs with at least a NE of the corresponding class. For example, if we have a question asking about a person, only paragraphs with a *person* entity will be validated.
- Coarse grained matching: since *location*, *organization* and *person* entities in English were grouped by the NER in a single category (this category is called *enamex*), we decided to group also questions asking about this kind of entities in a single category (*enamex* questions). Then, each of these questions can be answered with an entity of this category. For example, if a question asks about a location, a paragraph with a *organization* entity will be validated.

On the other hand, based on our experience, we grouped in both languages *count* and *time* questions into a category that can be answered by a *numeric* or *time* expression. We took this decision because the NER sometimes assigns the class *time* to *numeric* expressions and vice versa.

Named Entities Presence The validation process performed by this module follows the intuition that the NEs of a question must appear in a correct answer [4]. We could have applied this restriction in the retrieval phase (retrieving only paragraphs that contain these NEs), but we obtained better results when the restriction is applied at this step.

Only paragraphs that contain all the NEs of the question are validated by this module and returned as output. If a question does not have any NE, all the paragraphs are validated by this module because there are no evidences for rejecting them.

The restriction of containing the exact NE could seem very strict. In fact, it produced some errors in the last edition. We thought about using a relaxed version for allowing a little difference between NEs using the edit distance of Levenshtein [1]. However, we saw that this option produced false positives in NEs with a similar wording but that refer to different entities.

Since we were not sure about what matching was better, and taking into account the importance of NEs for supporting correctness, we decided to apply the strict version.

Acronym Checking This module is applied only in definition questions that ask for the meaning of an acronym (as for example *What is UNESCO?* or *What does UNESCO stand for?* Only paragraphs that are considered to contain a definition of the acronym are validated.

In order to apply this module, definition questions are analyzed to check whether they are asking about the meaning of an acronym. In that case, the acronym is extracted. Then, only paragraphs that contain the acronym inside a pair of brackets are validated in the current implementation of this module.

2.4 Selection of Final Answer

This module received the answers that accomplish the constraints checked in the previous step and decided the final answer for each question. In case of not having any candidate answer after the AV phase, the option NoA (what in ResPubliQA means that a system is not sure about finding a correct answer to a question and prefers not to answer it) is selected. NoA answers can receive the hypothetical answer that would be given in case of answering the question. These hypothetical answers are used for evaluating the validation performance. We gave in these cases the first answer according to the IR ranking.

If there is more than one paragraph at the end of the AV phase, we had two options for ranking answers and selecting the final one last year:

- The ranking given by the IR engine
- A ranking based on lemmas overlapping, with the possibility of including textual entailment (more details are given in [5]).

The ranking based on lemmas offered better results. However, we wanted to compare in this edition the pure IR system with the combination of IR and AV. Since we could only send two runs per language, we decided to use the IR ranking in both runs for a better comparison. Thus, we can study how the removal of paragraphs considered as incorrect affects the IR ranking.

3 Runs Submitted

The runs submitted were selected taking into account the objectives of our participation. These objectives were to study the improvement of the IR phase using more information about the question and its combination with a validation step.

We decided to submit two runs per language (we participated in English and Spanish) for the PS task: one run based only on the IR phase described in Section 2.2 and a second one that added the validation step (described in Section 2.3) to the output of the IR engine. More in detail, the submitted runs were as follows:

- **Spanish**
  - **Run 1:** the validation modules described in Section 2.3 (using the fine grained matching for the expected answer type) were applied to the output of the IR engine. If there was no paragraph after the validation process, the question was not answered (NoA option) and the first paragraph in the IR ranking was given as the hypothetical answer. In case of having more than a paragraph after the validation phase, the paragraph with the highest ranking according to the IR ranking among the validated paragraphs was given as answer.
  - **Run 2:** all the questions were answered using the first paragraph returned by the IR engine.
- **English**
  - **Run 1:** this run was similar to the Spanish first run except that it uses the coarse grained matching for the expected answer type.
  - **Run 2:** similar to the Spanish second run.

4 Results

The answers of each run were evaluated by human assessors and tagged as *correct* (R) or *incorrect* (W). The hypothetical answers given in case of choosing not to answer a question were evaluated as *unanswered* with a *correct* candidate answer (UR), or *unanswered* with an *incorrect* candidate answer (UI). The main evaluation measure was *c@1* (Formula (1)), while accuracy (Formula (2)) was used as a secondary evaluation measure.

$$c@1 = \frac{\#R}{n} + \frac{\#R}{n} * \frac{\#UR + \#UI}{n} \tag{1}$$

$$accuracy = \frac{\#R + \#UR}{n} \tag{2}$$

The results obtained by our system are shown in Table 1 for Spanish and Table 2 for English. These tables show also the validation performance of the system. This validation performance is calculated as the ratio of wrong hypothetical answers with respect to the whole amount of NoA answers. That is, if all the hypothetical answers were incorrect, the validation performance would be perfect.

Table 1. Results for Spanish runs.

Run	#R	#W	#UR	#UI	c@1	accuracy	validation performance
run 1	92	73	22	13	0.54	0.57	0.37
run 2	108	92	0	0	0.54	0.54	-

Table 2. Results for English runs.

Run	#R	#W	#UR	#UI	c@1	accuracy	validation performance
run 1	117	66	13	4	0.63	0.65	0.24
run 2	129	71	0	0	0.65	0.65	-

5 Analysis of results

According to Tables 1 and 2, our runs performed over 0.5 for both *accuracy* and *c@1*. However, we can see how the second run in each language, which did not include validation, performed better than the first one (second runs gave more correct answers). Therefore, the addition of the validation step reduced the performance of the system. These results were different to the ones obtained last year, where the validation phase improved the results.

The results of the two runs in each language were very similar according to *accuracy*, what is another indication of the low performance of validation. However, we have seen that the validation step allowed to remove some incorrect answers, contributing to return in the second runs some correct answers that were not given by the first runs. Therefore, the validation step can help in improving results, but it must reduce the amount of false negatives that it produces.

Most of the errors produced by the validation step were due to the NEs presence module. As it was already mentioned above, the criteria for deciding whether a NE appears in a paragraph can be too strict. This leads to errors discarding paragraphs, increasing the amount of false negatives given by the module. An example of these errors happened in question 13 (*What procedure does Mr. Sarkozy advocate concerning the internet?*), where the NE of the question was *Mr. Sarkozy*, and in some of the candidate paragraphs appeared *Mr Sarkozy* (without the dot after *Mr*). This simple change in the wording led to not answering to that question. Therefore, it is evident that we have to relax this constrain with the objective of reducing the amount of false negatives, while keeping the number of false positives.

One of the modifications included in our system this year was the use of different boost factors for different terms in the IR step. These different boost factors were given taking into account the NEs and focus of a question. This modification allowed a slight improvement in the performance of the IR engine and the overall results by increasing the ranking of correct answers.

Nevertheless, the IR engine has already a really good performance, and the question analysis output must be taken into account by more modules of the system if we want to obtain a higher improvement of the overall results.

We consider that better results could be obtained by improving the selection of the final answer. The improvement could be achieved by adding information from the question analysis and validation steps. In fact, the ranking based on lemmas that was used last year showed that additional information based on lemmas overlapping improved ranking.

In conclusion, the results showed that the IR performance is quite good. A better validation performance combined with more information for selecting the final answer is what our system needs for improving overall results.

6 Conclusions and Future Work

We have described in this paper our participation at ResPubliQA 2010. Our system has taken advantage of a powerful IR engine that has been slightly improved adding information from the question analysis.

Besides, a validation step was applied in order to remove possible incorrect answers from the pool of paragraphs returned by the IR engine. The validation has contributed to find more correct answers to some questions, but some of its components were too strict, removing also correct answers. Therefore, a relaxation of some of the constraints implemented in the validation step must be applied with the purpose of reducing the amount of false negatives without increasing the number of false positives.

On the other hand, the selection of the final answer was based only in the IR ranking after validation. A way of improving the overall performance would be to select the final answer taking into account also information of validation as well as the analysis of the question.

Acknowledgments

This work has been partially supported by the Spanish Ministry of Science and Innovation within the project QEAVis-Catiex (TIN2007-67581-C02-01), the Regional Government of Madrid under the Research Network MA2VICMR (S-2009/TIC-1542), the Education Council of the Regional Government of Madrid and the European Social Fund.

References

1. Vladimir Levensthein. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. In *Soviet Physics - Doklady*, volume 10, pages 707–710, 1966.
2. Joaquín Pérez-Iglesias, Guillermo Garrido, Álvaro Rodrigo, Lourdes Araujo, and Anselmo Peñas. Information Retrieval Baselines for the ResPubliQA Task. In *CLEF 2009. LNCS. To appear*, 2009.
3. Stephen Robertson and Steve Walker. Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval. In W. Bruce Croft and C. J. van Rijsbergen, editors, *SIGIR*, pages 232–241. ACM/Springer, 1994.
4. Álvaro Rodrigo, Anselmo Peñas, Jesús Herrera, and Felisa Verdejo. The Effect of Entity Recognition on Answer Validation. In *CLEF 2007, volume 5152 of Lecture Notes in Computer Science. Springer*, pages 483–489, 2007.

5. Álvaro Rodrigo, Joaquín Pérez, Anselmo Peñas, Guillermo Garrido, and Lourdes Araujo. Approaching Question Answering by means of Paragraph Validation. In *CLEF 2009. LNCS. To appear*, 2009.