Approaching Question Answering by means of Paragraph Validation

Álvaro Rodrigo, Joaquín Pérez, Anselmo Peñas, Guillermo Garrido, Lourdes Araujo
Dpto. Lenguajes y Sistemas Informáticos, UNED
{alvarory,joaquin.perez,anselmo,ggarrido,lurdes}@lsi.uned.es

Abstract

In this paper we describe the system we developed for taking part in monolingual Spanish and English tasks at ResPubliQA 2009. Our system was composed by an IR phase focused on improving QA results, a validation step for removing not promising paragraphs and a module based on ngrams overlapping for selecting the final answer. Furthermore, a selection module that used lexical entailment and ngram overlapping was developed in English. While the IR module has achieved very promising results, the performance of the validation module has to be improved. On the other hand, the ngram ranking improved the one given by the IR module and it worked better for English than for Spanish. Finally, the ranking was slightly improved when lexical entailment was used.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]; H.3.4 [**Systems and Software**]: [Question-answering (fact retrieval) systems]

Keywords

Question Answering, Answer Validation, Lexical Entailment

1 Introduction

The first participation of UNED at QA@CLEF (called this year ResPubliQA) is based on our experience as participants and organizers of the Answer Validation Exercise[1] (AVE) [11, 12, 17, 18, 19]. Our motivation for using Answer Validation (AV) in this task comes from the conclusions obtained in AVE, where it was shown that AV systems could contribute towards improving results in Question Answering (QA). In fact, some systems that took part in the last edition of QA@CLEF improved their results including an AV module [2, 20].

This year a paragraph containing a correct answer had to be returned for each question instead of the exact answer string of last editions. Since answer extraction was not necessary, we could concentrate on the validation of candidate paragraphs. Nevertheless, most of the experiments in AV have been performed with short answers but not with paragraphs containing an answer. Thus, checking the viability of performing paragraph validation in QA was one motivation for taking part in this task.

On the other hand, although according to the guidelines all the questions have an answer in the document collection, if a system is not sure about the correctness of an answer, the system can choose not to give any answer. In fact, the evaluation in this edition gives a higher reward

[1] http://nlp.uned.es/clef-qa/ave

for not giving an answer than for returning an incorrect one. Because of this modification in the evaluation we think it is important to carry out a validation step in order to decide if a correct answer can be returned or if no answer has been found. Thus, if our QA system is not sure about the correctness of all the candidate answers, no answer is returned.

In this paper we describe the main features of our QA system and the results obtained in monolingual English and Spanish. The rest of this paper is structured as follows: In Section 2 we describe the main components of our system. The description of the submitted runs is given in Section 3, while the results and their analysis are shown in Section 4. Finally, some conclusions and future work are given in Section 5.

2 System Overview

The main steps performed by our system are shown in Figure 1 and they are described in detail in the following subsections.

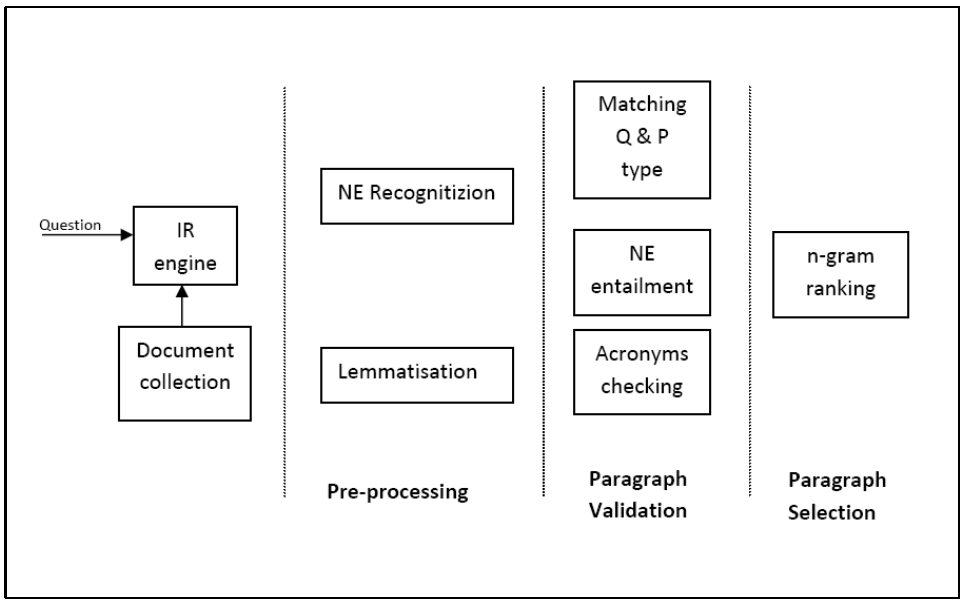

Figure 1: System's architecture.

2.1 Retrieval Phase

In this phase a first selection of paragraphs that are considered relevant for the proposed question are selected, therefore it will put focus on obtaining a first set of paragraphs ordered according to their relevance with the question. The precision in terms of retrieving the correct answer for the query within the top k[2] paragraphs, delimits in some sense the overall quality of the full system. In order to retrieve the most relevant paragraphs, the full collection has been indexed by paragraphs removing stopwords, specifically by language.

We used BM25 [16], which can be adapted to fit the specific characteristics of the data in use. More information about the selected retrieval model can be found in [13]. In this ranking function the effect of term frequency and document length to the final score of a document can be specified by setting up two parameters (b,k_1). The expression for BM25 ranking function[3] for a document d and query q is as follows:

[2]In the final system k has been fixed to 100

[3]An implementation of BM25 ranking function over Lucene can be found at :http://nlp.uned.es/ jperezi/Lucene-BM25/

$$R(q,d) = \sum_{t\ en\ q} \frac{freq_{t,d}}{k_1((1-b)+b\cdot\frac{l_d}{avl_d})+freq_{t,d}} \cdot \frac{N-df_t+0.5}{df_t+0.5} \quad (1)$$

Where N is the total number of documents in the collection; df_t is the number of documents in the collection that contain the term t; $freq_{t,d}$ is the frequency of the term t within document d; l_d is the length of the document and avl_d is the average length of documents within the collection. The values of the parameters should be fixed according to the data, as appears next:

- $b \in [0,1]$. Assigning 0 to b is equivalent to avoid the process of normalisation and therefore the document length will not affect the final score. If b takes 1, we will be carrying out a full normalisation $\frac{dl}{avdl}$.
- k_1, where $\infty > k_1 > 0$, allow us to control the effect of frequency in final score.

Retrieval Settings

For both languages stopwords have been removed and a stemming pre-process based on Snowball implementation of Porter algorithm has been applied. This implementation can be downloaded from http://snowball.tartarus.org/algorithms/. The stopwords lists applied can be found at http://members.unine.ch/jacques.savoy/clef/.

The BM25 parameters for both languages were fixed after a training phase with the English development data supplied by the organisation (No development Spanish data was released). These values are as next:

1. b: 0.6. Those paragraphs with a length over the average will obtain a slightly higher score.
2. k_1: 0.1. The effect of term frequency over final score will be minimised.

2.2 Pre-processing

In this step each question and each paragraph returned by the IR engine is pre-processed with the purpose of obtaining the following data:

- **Name Entities (NEs):** the Freeling NE recognizer [1] is applied in order to tag proper nouns, numeric expressions and temporal expressions for each question and each candidate paragraph. Besides, it is also included information regarding the type of the NE found. That is, for proper nouns we have types PERSON, ORGANIZATION and LOCATION (since Freeling does not supply this classification in English, these three types are grouped in the ENAMEX type when the QA system is used for English texts); NUMEX for numeric expressions and TIMEX for time expressions.
- **Lemmatisation:** the Freeling PoS tagger in Spanish and TreeTagger[4] in English are used for obtaining the lemmas of paragraphs and questions.

2.3 Paragraph Validation

This component receives as input the original questions and paragraphs as well as the pre-processed ones obtained in the previous step. The objective is to remove paragraphs that do not satisfy a set of constraints imposed by a question since, in that case, it is not likely to find a correct answer for this question in these paragraphs.

A set of modules for checking constraints have been implemented (3 in this edition, but more modules can be easily added to the system) and they are applied in a pipeline processing. That is, only paragraphs able to satisfy a certain constraint are checked against the following constraint. Finally, only paragraphs that satisfy all the implemented constraints are given to the following

[4] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/

step. In fact, it is possible to obtain no paragraph as output, what means that no paragraph is a candidate for containing a correct answer (according to this component). This situation changes the way of selecting the final answer as it is explained in Section 2.4.

The constraints implemented in this edition are explained in the following sections.

2.3.1 Expected Answer Type matching

Traditional QA systems typically apply as a first processing step an analysis of the input question where the expected answer type represents an important and useful information for the following steps [5, 9, 15]. Besides, there are some AV systems that use also information about the expected answer type with the purpose of checking whether the expected answer type matches the type of the answer to be validated [22, 10, 21, 6]. While some AV systems use this checking as a feature [21], others use it as a constraint that must be satisfied by an answer in order to be validated [22]. We think that a correct answer paragraph must contain at least one element whose type matches the expected answer type. This is why we decided to validate only paragraphs that contain elements of the expected answer type.

Firstly, the expected answer type is detected for each question. We based our taxonomy on the one used in the last editions of QA@CLEF. Thus, the defined types were: *count*, *time*, *location*, *organization*, *person*, *definition* and *other* (this is chosen when none of the previous categories is given). Then, although several machine learning methods have been successfully applied for question classification [8], given the small size of our taxonomy we decided to use the traditional approach based on hand-made patterns.

Secondly, for performing the matching process we took advantage of the fact that all the types in our taxonomy (except *definition* and *other*) match the possible NE types given by the pre-processing step. That is, *count* questions must be answered by numeric expressions, *time* questions must be answered by temporal expressions, etc. Then, the module validates paragraphs that contain at least a NE of the expected answer type and rejects the other paragraphs. In case of the expected answer type is *definition* or *other*, all the input paragraphs are validated because the system does not have enough evidences for rejecting them.

Our system can perform two kinds of expected answer type matching: the coarse grained matching and the fine grained matching. In the fine grained matching all the possible expected answer types and all the possible NE types are used. Thus, only paragraphs with at least one NE of the same type that the expected answer type will be validated. For example, if the expected answer type of a question is *person*, only paragraphs containing at least one NE of PERSON type will be validated.

However, in the coarse grained matching some types are grouped. That is, the expected answer types *organization*, *person* and *location* are grouped into a single one called *enamex*, which means that any NE of one of these types (PERSON, ORGANIZATION and LOCATION) can match with any *enamex* question. For example, if the expected answer type is *location* and the only NE in a paragraph is of type PERSON, the paragraph will be validated (while it would not be validated using the fine grained matching). In a similar way, *time* and *count* questions are grouped in a unique type and they can be answered by either numeric expressions or time expressions.

We decided to allow this double matching based on the intuition that NE types sometimes can be wrongly classified as for example the expression *in 1990*, which can be classified as a numeric expression when in fact can be also a temporal expression. Moreover, since the NE recognizer used in English did not give us a fine grained classification of *enamex* NEs, we needed to use the coarse grained matching in this language.

2.3.2 NE entailment

The validation process performed by this module follows the intuition that the NEs of a question are a so important information that they must appear in some way in the text that supports an answer [17]. This year the supporting snippet is the paragraph given as answer and following the previous intuition, the NEs of the question must appear in any answer paragraph.

This module receives as input the NEs of the question and the candidate paragraphs before being pre-processed (we explain the reason below). Then, only paragraphs that contain all the NEs of the question are validated by this module and returned as output.

The idea of containment used is a simple text matching of the NEs of the question in the paragraphs. It is not important if the matched element in the paragraph is or not a NE, because the important NEs are the ones of the question. In fact, this kind of matching allows to avoid possible errors in the recognition of NEs in the paragraphs. Another difference with the work performed in [17] is that we did not use the Levensthein distance [7] because in the development period it produced worse results.

If a question does not have any NE, then all the paragraphs are validated by this module because there are no evidences for rejecting them.

2.3.3 Acronym checking

This module works only over questions that ask about the meaning of a certain acronym as for example *What is NATO?* or *What does NATO stand for?*. The objective is to validate only paragraphs that could contain an explanation for these acronyms.

Firstly, the module checks whether the question is of *definition* type and whether it is asking about a word that only contains capitalized letters, which we called acronym. If the question satisfies these constraints, then the acronym is extracted.

Secondly, only paragraphs that can contain a possible definition for the extracted acronym are validated. In the current implementation it is considered that if a paragraph contains the acronym inside a pair of brackets then it might contain a definition of the acronym. For example, for question *What does ECSC stand for?*, where the acronym is *ECSC*, the paragraph in Figure 2 contains an explanation of the acronym and it would be validated by this module.

```
on the consequences of the expiry of the European Coal and Steel Community
      (ECSC) Treaty on international agreements concluded by the ECSC
```

Figure 2: An example of a paragraph containing the explanation to an acronym.

Similar to the other validation modules, if the restriction cannot be applied, that is, if the question is not asking about the definition of an acronym, all the input paragraphs are validated.

2.4 Paragraph Selection

Once all the restrictions have been applied, the system selects one paragraph among the ones validated in the previous step. Since AV has been successfully applied for performing the selection of answers in QA [21], it could be a natural option to use it since our system is already using an AV module. However, when an AV module is used for selection, it usually produces a confidence value that is considered for performing the selection. Since our AV module does not produce any confidence value, we decided to discard this option. Then, after some experiments performed at the development period we based the decision of which paragraph to select on the overlapping between question and answer paragraphs.

The paragraph selection works only when the validation process returns more than one candidate paragraph. If there is only one candidate paragraph, then it is the one selected. If there is no candidate paragraph, that means that no candidate paragraph was suitable for containing a correct answer. In these cases, it is considered that the system cannot find an answer and the system does not answer the question (an option that is allowed this year). Since in case of not giving any answer an hypothetical answer must be given for evaluation purposes, in this situation it is returned the paragraph that was chosen by the IR engine at the first position.

We have two modules for selecting the final answer: one based only on lemmas overlapping; and another one based on lemmas overlapping and Lexical Entailment.

2.4.1 Setting 1

As a way of avoiding different formulations of similar expressions we discarded stop words and measured overlapping using lemmas. Thus, the selection process is as follows:

1. Overlapping using 1-grams (lemmas) is measured. If the maximum overlapping with the question is achieved for only one paragraph, then that paragraph is selected. If the maximum overlapping is achieved for more than one paragraph, then the next step is performed.

2. The overlapping using 2-grams (lemmas) is measured over the paragraphs with the maximun overlapping using 1-grams. If the maximum overlapping with the question is achieved for only one paragraph, then that paragraph is selected. If the maximum overlapping is achieved for more than one paragraph, then the process is repeated with 3-grams, 4-grams and 5-grams stopping when there is still more than one paragraph with the maximun overlapping using 5-grams (lemmas) to perform the next step.

3. If there is more than one paragraph with the maximun overlapping using 5-grams (lemmas), then it is selected among the paragraphs with the maximum overlapping the one which obtains the higher ranking in the IR process.

2.4.2 Setting 2

Furthermore, for English we developed another version for this selection process that is based on Lexical Entailment. For this purpose we took advantage of a module based on WordNet relations and paths for checking the entailment between lexical units [3, 4]. The same process performed in setting 1 is applied, but there can be overlapping between a word in a paragraph and a word in a question if the two words are the same or the word in the paragraph entails (according to the entailment module based on WordNet) the word in the question. This new idea of overlapping is used with all the lengths of ngrams (from 1-grams to 5-grams).

3 Runs Submitted

In the first edition of ResPubliQA we took part in two monolingual tasks (English and Spanish), sending two runs for each of these tasks with the aim of checking different settings. All the runs applied the same IR process and the main differences are in the validation and selection steps. The characteristics of each run are as follows:

- **Monolingual English runs:** both runs applied for the validation process the coarse grained expected answer type matching (because with the NE recognizer used in English we can only perform this kind of matching), the NE entailment module and the acronym checking module. The differences come in the paragraph selection process:
  - **Run 1:** paragraph selection was performed by the module based on lemmas overlapping (setting 1) that was described in Section 2.4.1.
  - **Run 2:** paragraph selection was performed by the module based on lemmas overlapping and Lexical Entailment (setting 2) that was described in Section 2.4.2. The motivation for using this selection module was to study the effect of Lexical Entailment for ranking candidate answer paragraphs.

- **Monolingual Spanish runs:** in both runs the selection process was based on lemmas overlapping (setting 1 described in Section 2.4.1). Both runs applied the validation step in the same way for both the NE entailment module and the acronym checking module. The differences come in the use of the expected answer type matching module:
  - **Run 1:** it was applied the fine grained expected answer type matching.

– **Run 2:** it was applied the coarse grained expected answer type matching. The objective was to study the influence of using a fine grained or a coarse grained matching. It may be thought that the best option is the fine grained matching. However, possible errors in the classification given by the NE recognizer could contribute to obtain better results using the coarse grained option and we wanted to analyze it.

4 Analysis of the Results

The runs submitted to ResPubliQA 2009 were evaluated by human assessors who tagged each answer as *correct* (R) or *incorrect* (W). This year it was allowed to leave unanswered a question when there are no evidences about the correctness of the answer. In order to evaluate the performance of systems rejecting answers, the task allowed to return an hypothetical candidate answer when it was chosen not to answer a question. This answer could be the answer given if it was mandatory to answer all the questions. These answers were evaluated as *unanswered* with a *correct* candidate answer (UR), or *unanswered* with an *incorrect* candidate answer (UI). The main measure used for evaluation is c@1 (2). Moreover, accuracy (3) was also used as a secondary measure.

$$c@1 = \frac{\#R}{n} + \frac{\#R}{n} * \frac{\#UR + \#UI}{n} \tag{2}$$

$$accuracy = \frac{\#R + \#UR}{n} \tag{3}$$

The results obtained for the runs described in Section 3 are shown in Table 1 for English and Table 2 for Spanish. The results of a baseline system based only on the IR process described in Section 2.1 also appear in each Table. In this baseline, the answer given to each question was the first one according to the IR ranking.

Table 1: Results for English runs.

Run	#R	#W	#UR	#UI	accuracy	c@1
run 1	282	190	15	13	0.59	0.6
run 2	288	184	15	13	0.61	0.61
baseline	263	236	0	1	0.53	0.53

Table 2: Results for Spanish runs.

Run	#R	#W	#UR	#UI	accuracy	c@1
run 1	195	275	13	17	0.42	0.41
run 2	195	277	12	16	0.41	0.41
baseline	199	301	0	0	0.4	0.4

4.1 Results in English

Regarding English results, run 2 achieves a slightly higher amount of correct answers than run 1. Since the only difference between both runs was that run 2 used Lexical Entailment for ranking the candidate answers, the improvement was a consequence of using entailment. Although this is not a remarkable result for showing the utility of using entailment for ranking results in QA, it encourages us to explore more complex ways of using entailment for answer paragraphs ranking.

Comparing English runs with the English baseline it can be seen how the results of the submitted runs are about 10% better according to the given evaluation measures. A preliminary study

showed us that most of this variation in the results was a consequence of the different ways for ranking paragraphs and not the inclusion of the validation step. Then, the lemmas overlapping ranking used for the selection of paragraphs in the submitted runs has shown to be more appropriate for this task than the one based only on IR ranking when the QA system is working in English. Therefore, results suggest that it is useful to include information on lemmas when ranking the candidate paragraphs of a system.

4.2 Results in Spanish

The results of the Spanish submitted runs are quite similar as it can be seen in Table 2. Since the only difference between both runs was the expected answer type matching used, results suggest that there are no big differences between using one or another expected answer type matching. In fact, there were only 11 questions in which the given answers differ. In these 11 questions, 9 questions were incorrectly answered by both runs and in the other 2 ones, there was a correct answer for each run. Nevertheless, we detected that some of the errors obtained when the fine grained expected answer type matching was applied were caused by errors in the NE classification given by the NE recognizer. The possibility of having these errors was one of the motivations for using also coarse grained matching. However, when there was this kind of errors with the fine grained matching, the coarse grained matching did not help to find a right answer. Then, the preliminary analysis of the results show that the fine grained matching could contribute towards improving results, but it depends too much on the classification given by the used NE recognizer.

On the other hand, if we compare both submitted runs with the baseline run, we can see that the results according to the two evaluation measures are quite similar. This is different to the results obtained in English, in which the submitted runs performed better than the baseline. This means that the lemmas overlapping used for the selection process worked better in English than in Spanish. We want to perform a deeper analysis in order to study why there is such difference between the two languages.

4.3 Analysis of Validation

Given that one of our objectives for taking part at ResPubliQA was to study the impact of using validation, we find important to study the contribution of the validation modules in our QA system. Table 3 shows for each language the number of questions where each of the validation modules was applied. Despite the fact that the basic ideas of the modules where the same in both languages and the question set was also the same (the same questions but translated to each language), it can be seen in Table 3 how the numbers differ between languages. This was a consequence of different question formulations for each language and little variations in the implementation of modules for different languages. However, the number of questions that were leaved unanswered was almost the same in both languages as it can be seen in Tables 1 and 2.

Table 3: Number of question where each validation module was applied.

Language	Answer Type	NE entailment	Acronym
English	55	209	23
Spanish	44	179	6

Since the candidate answers given to unanswered questions were also evaluated, it can be measured the precision of systems validating answers (4). Table 4 shows the validation precision of the submitted runs for English and Spanish. In each language, the validation precision obtained was the same for both runs.

$$validation\ precision = \frac{\#UW}{\#UR + \#UW} \tag{4}$$

Table 4: Validation precision of the submitted runs in English and Spanish.

Language	Val. precision
English	0.46
Spanish	0.57

As it can be seen in Table 4, the validation precision achieved by the submitted runs is close to 50% (slightly better in Spanish and slightly worse in English). Therefore, the validation process applied by our QA system has not behaved very well.

We studied the errors produced by the validation process and we found that most of the errors were produced by the NE entailment module. On one hand, the constraint of having all the NEs of the question into the answer paragraph seemed to be very strict because a paragraph sometimes can omit some NEs that have been referred before in the document. Therefore, in the future we would like to study a way of relaxing these constraints that can allow us to improve results.

On the other hand, we found in Spanish some errors due to incorrect translations of the questions from English. For example, the NE *EEC* (which means European Economic Community) in question 17[5] was kept as *EEC* in Spanish, but the correct translation is *CEE* (which means *Comunidad Económica Europea*). This kind of errors in the translations caused that our system denied paragraphs that could contain correct answers.

Regarding the acronym checking, we found that its behaviour was quite good in Spanish but not in English. In fact, some questions were leaved unanswered in English because the acronym module was incorrectly applied. Therefore, we have to outperform this module in English.

Finally, the expected answer type matching was applied in a low amount of questions for both languages and we did not observe several problems in its performance. Now, we want to focus in improving its coverage so that it can help us in a higher amount of questions.

5 Conclusions and Future Work

In this paper we have described our QA system and the results obtained in both English and Spanish monolingual tasks at ResPubliQA. The main steps of our system were an IR phase focused on improving QA results, a validation step for rejecting no promising paragraphs and a selection of the final answer based on ngrams overlapping.

The IR ranking has provided a good performance obtaining better results in English than in Spanish, while the validation process was not very helpful. On the other hand, the ranking based on ngrams was able to improve results of the IR module in English, while it maintains the performance in Spanish. Besides, Lexical Entailment has shown to be informative for creating the answers ranking in English.

Future work is focused on solving the errors detected in each module, as well as developing modules for a broader range of questions. Furthermore, we want to perform a deeper study about the ranking of answers using ngrams in combination with Lexical Entailment.

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 TrebleCLEF Coordination Action, within FP7 of the European Commission, Theme ICT-1-4-1 Digital Libraries and Technology Enhanced Learning (Contract 215231), the Regional Government of Madrid under the Re-

[5]Why is it necessary to provide for information about certain foodstuffs in addition to those in Directive 79/112/EEC?

search Network MAVIR (S-0505/TIC-0267), the Education Council of the Regional Government of Madrid and the European Social Fund.

References

[1] Xavier Carreras, Isaac Chao, Lluís Padró, and Muntsa Padró. FreeLing: An Open-Source Suite of Language Analyzers. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC04). Lisbon, Portugal*, 2004.

[2] Sven Hartrumpf, Ingo Glöckner, and Johannes Leveling. University of Hagen at QA@CLEF 2008: Efficient Question Answering with Question Decomposition and Multiple Answer Streams. In *Working Notes for the CLEF 2008 Workshop, 17-19 September, Aarhus, Denmark*, 2008.

[3] Jesús Herrera, Anselmo Peñas, Álvaro Rodrigo, and Felisa Verdejo. UNED at PASCAL RTE-2 Challenge. In *Proceedings of the Second PASCAL Recognizing Textual Entailment Workshop*, April 2006.

[4] Jesús Herrera, Anselmo Peñas, and Felisa Verdejo. Textual Entailment Recognition Based on Dependency Analysis and *WordNet*. In Joaquin Quiñonero Candela, Ido Dagan, Bernardo Magnini, and Florence d'Alché Buc, editors, *MLCW*, volume 3944 of *Lecture Notes in Computer Science*, pages 231–239. Springer, 2005.

[5] Eduard Hovy, Laurie Gerber, Ulf Hermjakob, Michael Junk, and Chin-Yew Lin. Question Answering in Webclopedia. In *Proceedings of the Ninth Text REtrieval Conference, pages 655-664*, 2001.

[6] Adrian Iftene and Alexandra Balahur-Dobrescu. Answer Validation on English and Romanian Languages. In *LNCS. Springer Verlag. To appear*, 2009.

[7] Vladimir Levensthein. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. In *Soviet Physics - Doklady*, volume 10, pages 707–710, 1966.

[8] Xin Li and Dan Roth. Learning Question Classifiers. In *Proceedings 19th International conference on Computacional Linguistics*, 2002.

[9] Dan Moldovan, Sanda Harabagiu, Marius Pasca, Rada Mihalcea, Roxana Girju, Richard Goodrum, and Vasile Rus. The Structure and Performance of an Open-Domain Question Answering System. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 563–570, 2000.

[10] Véronique Moriceau, Xavier Tannier, Arnaud Grappy, and Brigitte Grau. Justification of Answers by Verification of Dependency Relations - The French AVE Task. In *Working Notes for the CLEF 2008 Workshop, 17-19 September, Aarhus, Denmark*, 2008.

[11] Anselmo Peñas, Álvaro Rodrigo, Valentín Sama, and Felisa Verdejo. Overview of the Answer Validation Exercise 2006. In Peters et al. [14], pages 257–264.

[12] Anselmo Peñas, Álvaro Rodrigo, and Felisa Verdejo. Overview of the Answer Validation Exercise 2007. In *Advances in Multilingual and Multimodal Information Retrieval, 8th Workshop of the Cross-Language Evaluation Forum, CLEF 2007, Budapest, Hungary, September 19-21, 2007, Revised Selected Papers, volume 5152 of Lecture Notes in Computer Science*, pages 237–248, 2007.

[13] Joaquín Pérez, Guillermo Garrido, Álvaro Rodrigo, Lourdes Araujo, and Anselmo Peñas. Information Retrieval Baselines for the ResPubliQA Task. In *this Volume*, 2009.

[14] Carol Peters, Paul Clough, Fredric C. Gey, Jussi Karlgren, Bernardo Magnini, Douglas W. Oard, Maarten de Rijke, and Maximilian Stempfhuber, editors. *Evaluation of Multilingual and Multi-modal Information Retrieval, 7th Workshop of the Cross-Language Evaluation Forum, CLEF 2006, Alicante, Spain, September 20-22, 2006, Revised Selected Papers*, volume 4730 of *Lecture Notes in Computer Science*. Springer, 2007.

[15] John Prager, Eric Brown, Anni Coden, and Dragomir R. Radev. Question-Answering by Predictive Annotation. In *Proceedings of the 23rd SIGIR Conference*, pages 184–191, 2000.

[16] 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.

[17] Álvaro Rodrigo, Anselmo Peñas, Jesús Herrera, and Felisa Verdejo. The Effect of Entity Recognition on Answer Validation. In Peters et al. [14], pages 483–489.

[18] Álvaro Rodrigo, Anselmo Peñas, and Felisa Verdejo. UNED at Answer Validation Exercise 2007. In Carol Peters, Valentin Jijkoun, Thomas Mandl, Henning Müller, Douglas W. Oard, Anselmo Peñas, Vivien Petras, and Diana Santos, editors, *CLEF*, volume 5152 of *Lecture Notes in Computer Science*, pages 404–409. Springer, 2007.

[19] Álvaro Rodrigo, Anselmo Peñas, and Felisa Verdejo. Overview of the Answer Validation Exercise 2008. In *LNCS. Springer Verlag. To appear*, 2009.

[20] Alberto Téllez-Valero, Antonio Juárez-González, Manuel Montes y Gómez, and Luis Villaseñor-Pineda. INAOE at QA@CLEF 2008: Evaluating Answer Validation in Spanish Question Answering. In *Working Notes for the CLEF 2008 Workshop, 17-19 September, Aarhus, Denmark*, 2008.

[21] Alberto Téllez-Valero, Antonio Juárez-González, Manuel Montes y Gómez, and Luis Villaseñor-Pineda. Using Non-Overlap Features for Supervised Answer Validation. In *LNCS. Springer Verlag. To appear*, 2009.

[22] Rui Wang and Günter Neumann. Information Synthesis for Answer Validation. In *LNCS. Springer Verlag. To appear*, 2009.