Expert Systems with Applications 40 (2013) 5811–5816

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Answering questions about European legislation

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

Dpto. Lenguajes y Sistemas Informáticos, UNED, Spain

ARTICLE INFO

Keywords:
Question Answering
Answer Validation

ABSTRACT

ResPubliQA is a Question Answering (QA) evaluation task over European legislation whose first edition was proposed at the Cross Language Evaluation Forum (CLEF) 2009. The exercise consists of extracting a relevant paragraph of text that satisfies the information need expressed by a natural language question. The definition of the task allows to compare current QA technologies with pure Information Retrieval (IR) approaches and to introduce Answer Validation technologies in QA systems. In this paper we describe a system developed for this task. Our system is composed by an IR phase focused on improving QA results, a validation step for removing not promising paragraphs and a module based on n-grams overlapping for selecting the final answer, as well as a selection module that uses Lexical Entailment. While the IR module has contributed to obtain promising results, the performance of the validation module has to be improved. On the other hand, the n-gram ranking improved the results of the ranking given by the IR module.

1. Introduction

Question Answering (QA) systems receive a question in natural language and return small snippets of text that contain an answer to the question (Voorhees & Tice, 1999). QA systems have lots of applications, as for example to make easier the access of nonexpert users to the great amount of information available in digital libraries.

ResPubliQA is a QA evaluation task that proposed at the Cross Language Evaluation Forum (CLEF) 2009 (Peñas et al., 2010) and 2010 (Forner et al., 2010). This exercise is the continuation of the QA campaigns celebrated at CLEF from 2003 (Magnini et al., 2003). The task proposed that given a pool of 500 independent questions in natural language about European legislation, systems had to return a passage – not a exact response – that answers each question.

In ResPubliQA, if a system was not sure of finding a correct answer to a question, the system could choose not to give any answer to that question. In fact, the evaluation gave a higher reward for not giving an answer than for returning an incorrect one.

We proposed a system that in case of not being sure about the correctness of the candidate answers to a question, it does not return any answer to that question.

In this paper we describe the main features of ResPubliQA and a proposal of system for this task. The paper is organized as follows: Section 2 describes the evaluation performed at ResPubliQA, while our proposal is described in Section 3. The configuration of the experiments tested with the ResPubliQA evaluation collection is given in Section 4, while the results are shown in Section 5. Finally, some conclusions and future work are given in Section 6.

2. ResPubliQA

The ResPubliQA exercise was aimed at retrieving answers to a set of 500 questions. The answer of a question was a paragraph that satisfies completely the information need expressed by the question. The hypothetical user considered for this exercise is a person interested in making inquiries in the law domain, specifically on the European legislation. The ResPubliQA document collection is a subset of JRC-Acquis,[1] a corpus of European legislation, freely available, that has parallel translations aligned at document level in many European languages. It comprises contents, principles and political objectives of the EU treaties; the EU legislation; declarations and resolutions; international agreements; as well as acts and common objectives. Texts cover various subject domains, including economy, health, information technology, law, agriculture, food, politics and more. This collection of legislative documents currently includes selected texts written between 1950 and 2006 with parallel translations in 22 languages.

The exercise was proposed in Basque, Bulgarian, English, French, German, Italian, Portuguese, Romanian and Spanish. Participants at ResPubliQA were allowed to submit just one response per

* Corresponding author. Tel.: +34 91 398 96 93.
E-mail addresses: alvarory@lsi.uned.es (Álvaro Rodrigo), joaquin.perez@lsi.uned.es (J. Pérez-Iglesias), anselmo@lsi.uned.es (A. Peñas), ggarrido@lsi.uned.es (G. Garrido), lurdes@lsi.uned.es (L. Araujo).

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

0957-4174/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2013.05.008

question and up to two runs per task. Each question had to receive one of the following responses:

1. A paragraph with the candidate answer.
2. The string NOA to indicate that the system preferred not to answer the question.

Optionally, systems that preferred to leave some questions unanswered could decide to submit also a candidate paragraph to that questions. This option was used to evaluate the validation performance.

3. Overview of the system

QA systems typically employ a pipeline approach in which the sequence of steps is: question analysis, document retrieval, passage selection and answer extraction (Hovy, Gerber, Hermjakob, Junk, & Lin, 2001; Moldovan et al., 2000). Our system is based in this classical architecture and it works in English and Spanish. However, since answer extraction was not necessary in ResPubliQA, we did not include this step.

On the other hand, we included a validation step given that the evaluation in ResPubliQA gives a higher reward for not giving an answer than for returning an incorrect one. Thus, if our system is not sure about the correctness of all the candidate answers to a question, no answer is returned to that question.

The main steps performed by our system are described in detail in the following subsections.

3.1. Retrieval phase

A selection of paragraphs considered relevant for the question are selected in this phase, sorting the retrieved paragraphs according to their relevance to the question. For this purpose, the full collection has been indexed by paragraphs removing stopwords and performing a stemming pre-process.

The selection of an adequate retrieval model is a key part of the task. By applying an inadequate retrieval function, a subset of candidate paragraphs where the answer cannot appear would be returned and thus, any subsequent technique applied to detect the answer within this subset will fail.

In general, retrieval models are built around three basic statistics from the data: frequency of terms in a document; frequency of a term in the collection, where document frequency (DF) or collection frequency (CF) can be used; and document length. The ideal ranking function for this task should be adaptable enough to fit the specific characteristics of the data. For the ResPubliQA task, documents are actually paragraphs with an average length of 10 terms, and the frequency of question terms within a paragraph hardly exceeds one. A good candidate paragraph for containing the answer of a question is one that has the maximum number of question terms (excluding stopwords) and has a length similar to the average (to avoid giving too much importance to term frequency within the paragraph).

The use of the classic Vector Space Model (VSM) (Salton, Wong, & Yang, 1975) is not an adequate option for this task because this model typically normalises the weight assigned to a document with the document length. This causes that those paragraphs that contain at least one question term and have the lowest length will obtain the highest score. Moreover, the typical saturation of terms frequency used in this model, applying logarithm or root square, gives too much relevance to the term's frequency.

We used an Information Retrieval approach based on the Okapi-BM25 (Robertson & Walker, 1994) ranking function, which can be adapted to fit the specific characteristics of the data in use. The effect of term frequency and document length to the final score of a document can be specified in this ranking function by setting up two parameters (b,k_1). The expression for BM25 ranking function for a document d and query q is shown in Formula (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 must be fixed according to the data taking into account:

- $b \in [0,1]$. To assign 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 $\infty > k_1 > 0$, allows us to control the effect of frequency in final score.

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

The BM25 parameters of our system were fixed after a training phase with the English development data supplied by the organisation. These values were:

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.

3.2. Pre-processing

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 (Carreras, Chao, Padró, & Padró, 2004) is applied in order to tag proper nouns, numeric expressions and temporal expressions for each question and paragraph. Besides, it is also included information about the type of each NE. 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[2] in English are used for obtaining the lemmas of paragraphs and questions.

3.3. Paragraph Validation

This component receives 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 that question in those paragraphs.

A set of modules for checking constraints have been implemented. These modules 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 constraints are given to the following step. In fact, it is possible to obtain no paragraph as output, what means that no

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

paragraph was consider a candidate for containing a correct answer (according to this component).

The constraints implemented are explained in the following sections.

3.3.1. Expected answer type matching

QA systems typically apply an analysis of the input question, where the expected answer type represents an important and useful information for the following steps (Prager, Brown, Coden, & Radev, 2000). 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 previous editions of QA@CLEF. Thus, we used the types: *count*, *time*, *location*, *organization*, *person*, *definition* and *other*. Although several machine learning methods have been successfully applied for question classification (Li & Roth, 2002), given the small size of our taxonomy we decided to use a traditional approach based on hand-made patterns.

Secondly, we took advantage of the fact that 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. 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: a coarse grained matching and a fine grained matching. All the possible expected answer types and all the possible NE types (PERSON, ORGANIZATION, LOCATION, NUMEX and TIMEX) are used in the fine grained matching. Thus, only paragraphs containing at least one NE of the same type as the expected answer type will be validated. For example, if the expected answer type of a question is *person*, only paragraphs containing at least a NE of PERSON type will be validated.

However, some types are grouped in the coarse grained matching. In fact, the expected answer types *organization*, *person* and *location* are grouped into a single one called *enamex*. This 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 or time expressions.

We decided to allow this double matching based on the intuition that NE sometimes can be wrongly classified, as for example the expression *in 1990*, which can be classified as a numeric expression when in fact it is a temporal expression. Moreover, since the NE recognizer used in English did not give us a fine grained classification of *enamex* NEs (there is no difference among PERSON, ORGANIZATION and LOCATION), we needed to use the coarse grained matching in this language.

3.3.2. NE entailment

The validation process performed by this module follows the intuition that the NEs of a question are an important information that must appear in some way in the text that supports an answer (Rodrigo, Peñas, Herrera, & Verdejo, 2007). Since the supporting snippet in ResPuliQA is the paragraph given as answer (that is, the paragraph is both answer and supporting snippet), we considered that the NEs of the question must appear in any answer paragraph. If a question does not have any NE, all the paragraphs are validated by this module because there are no evidences for rejecting them.

This module receives as input the NEs of the question, as well as the candidate paragraphs before being pre-processed. Then, only paragraphs that contain all the NEs of the question are validated 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 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.

3.3.3. 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 of this module is to validate 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, the acronym is extracted.

Secondly, only paragraphs that can contain a possible definition to 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 Fig. 1 contains an explanation of the acronym and it would be validated by this module.

Similar to the other validation modules of the system, 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.

3.4. Paragraph selection

Once all the restrictions have been applied, the system selects a paragraph among the ones validated in the previous step. 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 a 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

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

Fig. 1. An example of a paragraph containing the explanation to an acronym.

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 in ResPubliQA). Since in case of not giving any answer, an hypothetical answer can be given, our system returns in this situation the paragraph in the first position of the IR ranking.

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. Both modules are explained below.

3.4.1. Setting 1

We decided to discard stop words and measure overlapping using lemmas in order to avoid different formulations of similar expressions. 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 more than one paragraph achieves the maximum overlapping, 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 by only one paragraph, then that paragraph is selected. If the maximum overlapping is achieved by more than one paragraph, 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), the paragraph that has the higher ranking in the IR process among the ones with maximum overlapping is selected.

3.4.2. Setting 2

We developed in English another version for the 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. 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 idea of overlapping is used with all the lengths of n-grams (from 1-grams to 5-grams).

4. Experimental setting

We performed different experiments over the test collection of ResPubliQA 2009 in order to test our system in English and Spanish. In addition, we compared the results of a pure IR system, as the one described in Section 3.1, with our system.

4.1. Proposed Experiments

We evaluate our approach both in English and Spanish, testing two runs in each language. Moreover, we test in each language a run (called baseline) that selects as the final answer the paragraph in the first position of the IR ranking. The characteristics of each run were as follows:

- **Monolingual English runs:** both runs applied in the validation process the coarse grained expected answer type matching (because the NE recognizer used in English allowed to perform only this kind of matching), the NE entailment module and the acronym checking module. The differences came in the paragraph selection process:
  - **Run 1:** paragraph selection was performed by the module based on lemmas overlapping described in Section 3.4.1.
  - **Run 2:** paragraph selection was performed by the module based on lemmas overlapping and Lexical Entailment described in Section 3.4.2. The motivation for using this selection module was to study the effect of Lexical Entailment for ranking answer paragraphs.
- **Monolingual Spanish runs:** in both runs the selection process was based on lemmas overlapping (setting 1 described in Section 3.4.1). Both runs applied the validation step in the same way for both the NE entailment module and the acronym checking module. The differences came 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.

5. Results

Runs submitted to ResPubliQA were evaluated by human assessors, who tagged each answer as *correct* (*R*) or *incorrect* (*W*). 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.These questions were evaluated as *unanswered* with a *correct* candidate answer (UR), or *unanswered* with an *incorrect* candidate answer (UI). The main measure used for evaluation was c@1 (Formula (2)) (Peñas & Rodrigo, 2011). Moreover, accuracy (Formula (3)) was also used as a secondary measure, considering that all the questions were answered.

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

$$\text{Accuracy} = \frac{\#R + \#UR}{n} \quad (3)$$

The results obtained over the English test collection are shown in Table 1, while the results in Spanish are shown in Table 2, which contains also the results of the best QA system in Spanish in ResPubliQA 2009 (the best result in English was obtained by our run 2).

5.1. Results in English

The results in English show that run 2 achieves a slightly higher amount of correct answers than run 1 (a not significant difference). Since the only difference between both runs was the fact 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

Table 1
Results for English runs.

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

Table 2
Results for Spanish runs.

Run	#R	#W	#UR	#UI	Accuracy	c@1
Best System	218	248	0	34	0.44	0.47
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

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 study showed us that most of this variation in the results was a consequence of the different ways for ranking paragraphs. Therefore, the lemmas overlapping ranking used for the selection of paragraphs 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. These results suggest that it is useful to include information on lemmas when ranking the candidate paragraphs of a system.

5.2. 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, results suggest that there are no big differences between using one or another expected answer type matching. We detected that some of the errors produced by the fine grained expected answer type matching were caused by errors in the classification given by the NE recognizer. The possibility of having these errors was one of the motivations for using also coarse grained matching. However, the coarse grained matching did not help to find a right answer when there was an error of the fine grained matching. Then, the analysis of the results showed that the fine grained matching could contribute towards improving results, but it depends too much on the classification given by the NE recognizer.

On the other hand, if we compare both 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, where 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.

5.3. Analysis of validation

We found important to study the contribution of the validation step in our QA system given that ResPubliQA took into account this process. 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 each module for each language. However, the number of questions that were left unanswered was almost the same in both languages (as it can be seen in Tables 1 and 2).

Furthermore, it can be measured the precision of systems validating answers (Eq. (4)) taking into account that the candidate answers given to unanswered questions were also evaluated. Table 4 shows the validation precision of the different runs for English and Spanish, where it can be seen how the validation precision was the same for the runs of the same language. As it can be seen in the Table, the validation precision is close to 50% (slightly better in Spanish and slightly worse in English).

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

We studied the errors produced by the validation process and found that most of them were produced by the NE entailment module. The constraint of having all the NEs of the question into the answer paragraph seemed to be very strict. We observed that 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 this constraint in order to improve results.

Regarding acronym checking, we found that its behaviour was quite good in Spanish but not in English. In fact, some questions were left unanswered in English because the acronym module was applied to questions that do not contain an acronym.

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

6. Conclusions and future work

In this paper we have described a Question Answering system and the results obtained in English and Spanish over the ResPubliQA test collection, which is focused on European legislation. The main steps of our system were an Information Retrieval phase focused on improving Question Answering results, a validation step for rejecting no promising paragraphs and a selection of the final answer based on n-grams overlapping.

The Information Retrieval 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 n-grams was able to improve results of the Information Retrieval 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. We want to improve the performance of validation by including deeper semantic analysis, such as semantic role labelling, and more complex Textual Entailment techniques. Furthermore, we would like to perform a broader study about the ranking of answers using n-grams in combination with Lexical Entailment.

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

Table 4
Validation precision of the submitted runs.

Language	Val. precision
English	0.46
Spanish	0.57

Acknowledgments

This work has been partially supported by the Spanish Ministry of Science and Innovation, through the project Holopedia (TIN2010-21128-C02) and the Regional Government of Madrid, through the project MA2VICMR (S2009/TIC1542)

References

Carreras, X., Chao, I., Padró, L., & Padró, M. (2004). FreeLing: An open-source suite of language analyzers. In *Proceedings of the fourth international conference on Language Resources and Evaluation (LREC04)*. Lisbon, Portugal.

Forner, P., Giampiccolo, D., Magnini, B., Peñas, A., Rodrigo, Á., & Sutcliffe, R. F. E. (2010). Evaluating multilingual question answering systems at CLEF. In N. Calzolari, K. Choukri, B. Maegaard, J. Mariani, J. Odijk, & S. Piperidis, et al. (Eds.), *LREC*. European Language Resources Association.

Hovy, E., Gerber, L., Hermjakob, U., Junk, M., & Lin, C.-Y. (2001). Question answering in webclopedia. In *Proceedings of the ninth text retrieval conference* (pp. 655–664).

Li, X., & Roth, D. (2002). Learning question classifiers. In *Proceedings 19th international conference on computacional linguistics*.

Magnini, B., Romagnoli, S., Vallin, A., Herrera, J., Peñas, A., Peinado, V., et al. (2003). The multiple language question answering track at CLEF 2003. In C. Peters, J. Gonzalo, M. Braschler, & M. Kluck (Eds.), *CLEF* (Vol. 3237, pp. 471–486). Springer, Lecture Notes in Computer Science.

Moldovan, D., Harabagiu, S., Pasca, M., Mihalcea, R., Girju, R., Goodrum, R., & Rus, V. (2000). The structure and performance of an open-domain question answering system. In *Proceedings of the 39th annual meeting of the association for computational linguistics* (pp. 563–570).

Peñas, A., Forner, P., Sutcliffe, R., Rodrigo, Á., Forascu, C., Alegria, I., Giampiccolo, D., Moreau, N., & Osenova., P. (2010). Overview of ResPubliQA 2009: question answering evaluation over european legislation. In *CLEF 2009, LNCS, to appear.*

Peñas, A., & Rodrigo, Á. (2011). A simple measure to assess non-response. In *ACL* (pp. 1415–1424). The Association for Computer Linguistics.

Prager, J., Brown, E., Coden, A., & Radev, D.R. (2000). Question-answering by predictive annotation. In *Proceedings of the 23rd SIGIR conference* (pp. 184–191).

Robertson, S., & Walker, S. (1994). Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval. In W. B. Croft & C. J. van Rijsbergen (Eds.), *SIGIR* (pp. 232–241). ACM/Springer.

Rodrigo, Á., Peñas, A., Herrera, J., & Verdejo, F. (2007). The effect of entity recognition on answer validation. In C. Peters, P. Clough, F. C. Gey, J. Karlgren, B. Magnini, & D. W. Oard, et al. (Eds.), *CLEF. Lecture Notes in Computer Science* (Vol. 4730, pp. 483–489). Springer.

Salton, G., Wong, A., & Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM, 18*, 613–620.

Voorhees, E.M., & Tice, D.M. (1999). The TREC-8 question answering track evaluation. In *Text retrieval conference TREC-8* (pp. 83–105).