Improving Question Answering by Combining Multiple Systems via Answer Validation

Alberto Téllez-Valero[1], Manuel Montes-y-Gómez[1],
Luis Villaseñor-Pineda[1], and Anselmo Peñas[2]

[1] Instituto Nacional de Astrofísica, Óptica y Electrónica
Grupo de Tecnologías del Lenguaje
Luis Enrrique Erro no. 1, Sta. María Tonantzintla, Pue.; 72840; Mexico
{albertotellezv,mmontesg,villasen}@inaoep.mx
[2] Universidad Nacional de Educación a Distancia
Depto. Lenguajes y Sistemas Informáticos
Juan del Rosal, 16; 28040; Spain
anselmo@lsi.uned.es

Abstract. Nowadays there exist several kinds of question answering systems. According to recent evaluation results, most of these systems are complementary (i.e., each one is better than the others in answering some specific type of questions). This fact indicates that a pertinent combination of various systems may allow improving the best individual result. This paper focuses on this problem. It proposes using an answer validation method to handle this combination. The main advantage of this approach is that it does not rely on internal system's features nor depend on external answer's redundancies. Experimental results confirm the appropriateness of our proposal. They mainly show that it outperforms individual system's results as well as the precision obtained by a redundancy-based combination strategy.

1 Introduction

Question Answering (QA) systems are a kind of search engines that allow responding to questions written in unrestricted natural language. Different to traditional IR systems that focus on finding relevant documents for general user queries, this kind of systems are especially suited to resolve very specific information needs.

Currently, given the great number of its potential applications, QA has become a promising research field. As a result, several QA methods have been developed and different evaluation forums have emerged (such as those at TREC[3] and CLEF[4]). Latest results from these forums evidenced two important facts

[3] Text REtrieval Conference. http://trec.nist.gov/

[4] Cross Language Evaluation Forum. http://www.clef-campaign.org/

about the state of the art in QA. On the one hand, they indicated that it already does not exist any method capable of answering all types of questions with similar precision rates. On the other hand, they also revealed that most current QA systems are complementary. That is, each system tends to be better than the others in answering some specific type of questions. Just as an example, in the Spanish QA evaluation at CLEF 2005, the best individual QA system could only answer 42.5% of the questions, whereas the ideal combination of correct answers from all participating systems could achieved a precision of 73.5% [1]. Based on these two facts, a new problem has emerged, namely, how to automatically get the appropriate combination of answers from several QA systems.

This paper focuses on this new problem. It proposes using an answer validation method to handle a superficial combination of several QA systems. It is important to mention that answer validation was mainly conceived as a means to help individual QA systems to automatically detect its own errors [2]. In accordance with this idea, several QA systems have included an answer validation module that helps them in deciding whether a candidate answer should be accepted or rejected [3]. Our proposal goes a step forward demonstrating the usefulness of answer validation for combining several complementary QA systems. In other words, this paper shows the effectiveness of answer validation for leading an ensemble of QA systems.

The rest of the paper is organized as follows. Section 2 describes some related work on QA ensemble approaches. Section 3 presents our general proposal about using answer validation as integration mechanism for combining several QA systems. Section 4 gives some details on the answer validation method. Section 5 shows some evaluation results in Spanish QA. Finally, section 6 offers some conclusions and ideas for the future work.

2 Related Work

Ensemble methods are very popular in machine learning tasks. They are based on the idea of using multiple classifiers to solve a common problem [4]. The success of these methods has motivated the implementation of "ensemble" approaches for other tasks. In particular, in question answering, the objective of an ensemble method is to combine the capacities of several QA systems in order to increase the number of correct answers.

Ensemble methods for QA are of two main types: internal and external. In the internal ensembles the combination of systems occurs at component level. Traditionally a QA system has three main components: one for question analysis, one for passage retrieval, and another one for answer extraction. Therefore, this kind of ensembles distinguishes for applying more than one technique in some particular component. For instance, [5] describes a QA system that uses several passage retrieval methods, and [6] presents a system that applies two distinct strategies at each component.

On the other hand, external or superficial ensembles combine different QA systems at answer level, i.e., they directly combine the answers extracted by several systems and select one of them as final answer. In this case, it is possible to distinguish two different combination strategies. The first one is solely based on answer's redundancies, i.e., the ensemble selects as final answer the most frequent one [7]. The second one, in contrast, not only takes into account the answer's redundancies but also a confidence value associated with the capability of each system to answer each specific type of question [8]. It is also important to mention that there are some ensemble methods for multilingual QA [9]. These methods consider answers from different languages and select the final answer based on its monolingual ranking as well as on its multilingual redundancy.

3 Proposed Ensemble Architecture

Figure 1 shows the general scheme of our proposal for a QA ensemble. This ensemble uses an answer validation method to superficially combine several QA systems.

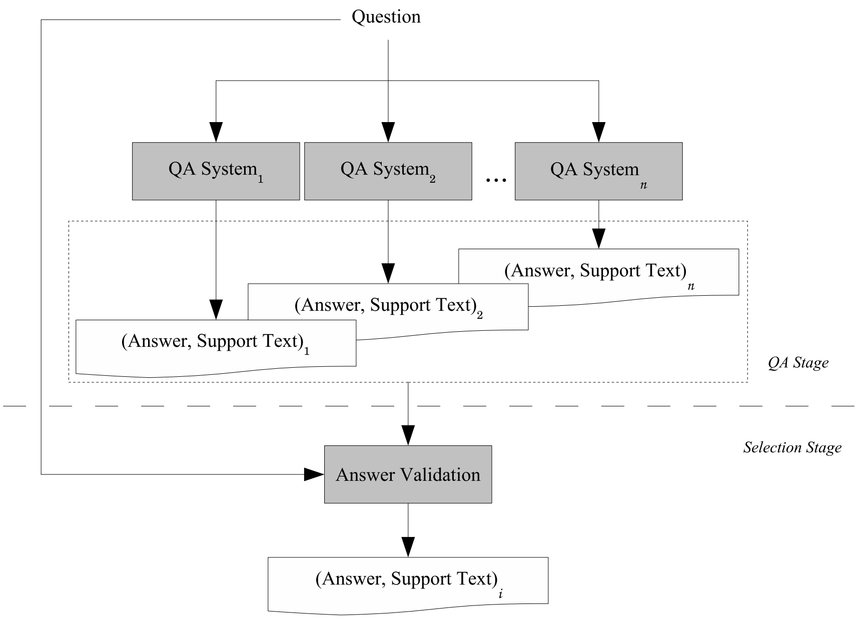
Question
QA System$_1$
QA System$_2$
...
QA System$_n$
(Answer, Support Text)$_n$
(Answer, Support Text)$_2$
(Answer, Support Text)$_1$
QA Stage
Selection Stage
Answer Validation
(Answer, Support Text)$_i$

Fig. 1. QA ensemble based on answer validation

Our QA ensemble consists of two main stages. In the first stage (called QA stage), several different QA systems extract – in parallel – a candidate answer (with its corresponding support text) for the given question. Then, in the second stage (called selection stage), an answer validation module evaluates – one by one – the candidate answers and selects as output the first accepted answer (for details on the validation process refer to section 4). In the case all candidate answers were rejected the output is set to NIL.

Given that the answer validation method is not perfect, the order of evaluation of the candidate answers is very relevant. Our current implementation considers a random order as well as a decreasing order based on the general confidence (accuracy) of the used QA systems.

The proposed ensemble distinguishes from previous approaches in three main concerns. First, it does not require to know (or adjust) internal details of the participating QA systems. Second, different to other previous external ensembles, it does not dependent on answer's redundancies. This is of crucial importance since there are many questions for which only one or very few QA systems could extract the correct answer. Third, in the case that there is not any correct answer, our approach could return a NIL answer, i.e., it is not obligated (as others) to always select one candidate answer. Finally, given the use of an answer-validation selection strategy, our ensemble not only returns correct but also supported answers.

4 Answer Validation Module

Given a question, a candidate answer and a support text, the answer validation module must decide whether to accept or reject the candidate answer. In other words, it must determine if the specified answer is correct and supported [2].

Our answer validation module is based on the idea of recognizing the textual entailment between the support text (T) and an affirmative sentence (called hypothesis, H) created from the combination of the question and the answer. The entailment between the pair (T, H) occurs when the meaning of H can be inferred from the meaning of T [10].

Figure 2 shows the general architecture of the answer validation module. As it can be seen, this module is based on a supervised learning approach and considers three main processes: hypothesis generation, feature extraction and entailment recognition.

4.1 Hypothesis Generation

The main task of this initial process is to construct two distinct hypotheses combining the given question and answer. In order to do that it firstly applies a superficial syntactic analysis over the question[5]. Then, using the obtained syntactic tree, it generates both hypotheses.

The first hypothesis (H) is constructed by replacing the nominal phrase that contains the interrogative particle by the given answer. For instance, given the question *"How many inhabitants are there in Longyearbyen?"* and the answer

[5] The language analysis used in the answer validation module was carry out with the open source tool called Freeling [11].

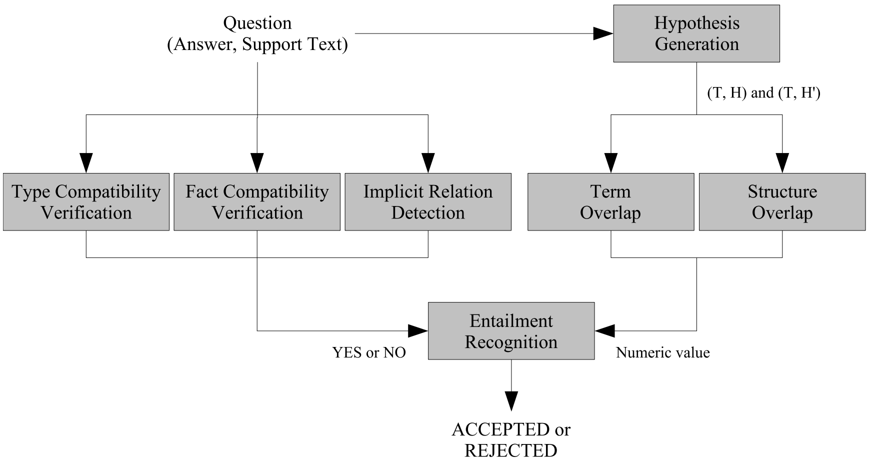

Fig. 2. General architecture of the answer validation module

"180 millions of inhabitants", this approach allows generating the hypothesis *H="180 millions of inhabitants are there in Longyearbyen"*.

The second hypothesis (*H'*) is obtained doing a simple transformation on *H*. The idea is to detect the main verb phrase of the H (that is the main verb phrase of the question) and then interchange its surrounding nominal phrases. This way the second hypothesis for our example is *H'="in Longyearbyen are there 180 millions of inhabitants"*.

4.2 Feature Extraction

We used two different kinds of features for the entailment recognition. On the one hand, some features that indicate the compatibility of question and answer. On the other hand, some classical textual entailment features that denote the level of similarity between the support text (T) and the generated hypotheses *H* or *H'*. The following subsections describe all these features.

Type Compatibility Verification. This process captures the situation where the semantic class of the evaluated answer does not correspond to the expected class of answer (in accordance with the given question). For instance, having the answer *"yesterday"* for the question *"How many inhabitants are there in Longyearbyen?"*.

In essence, this process calculates a boolean value that indicates if the general-class restriction is satisfied. This restriction is TRUE if the semantic class of the candidate answer and the expected class of the answer are equal; in other case, it is set to FALSE.

In the current module's implementation, three general classes are considered: quantities, dates, and names. Moreover, the question classification (i.e., the definition of the expected class of the answer) is done using the KNN supervised algorithm with $K = 1$ and the answer classification is done by a name entity recognition method.

Fact Compatibility Verification. This process focuses on the situation where the question asks about a specific fact and the answer makes reference to another different fact. For instance, answering *"eight"* to the example question, using as support text *"... when eight animals parade by the principal street in Longyearbyen, a town of a thousand of inhabitants"*.

With the aim of capturing this situation, this process determines a boolean value that indicates if a specific-type restriction is satisfied. In order to determine the specific target fact concerning the question it is necessary to perform the following procedure: (*i*) construct the syntactic tree of the question, and (*ii*) extract the principal noun from the noun phrase that contains the interrogative particle. Applying this procedure over the example question, the word *"inhabitants"* was selected as the specific target fact.

Once extracted the specific target fact from the question, it is possible to evaluate the specific-type answer restriction. Its value is set to TRUE if the specific target fact happens in the support text, in the immediate answer context (one content word to the right or left). In any other case its value is set to FALSE. Therefore, the candidate answer *"eight"* has its value set to FALSE since its immediate context (*"eight animals"*) does not contains the noun *"inhabitants"*. On the contrary, the candidate answer *"thousand"* will be have its value set to TRUE, since the noun *"inhabitants"* occurs in its immediate context (*"town thousand inhabitants"*).

It is important to notice that not for all questions it is possible to establish a specific target fact (e.g., consider the question *"When was Amintore Fanfani born?"*). In these cases we considered – by default – that all candidate answers satisfied the specific-type restriction.

Implicit Relation Detection. Commonly, support texts present language phenomena such as apposition and adjectival phrases. This kind of phenomena makes implicit a relation between some elements (noun phrases) from the support text, and therefore, causes a detriment in the overlap between T and H. For instance, in the text *"the quinua, an American cereal of great nutritional value,"*, the verb *"is"* is implicit, it according to the hypotheses generates from the question and answer shows in the table 1.

In order to help the entailment recognition process to adequately treat these cases, we decide including a Boolean feature that simply indicates the existence of implicit information, i.e., the presence of some apposition or adjectival phrase.

The detection of this language phenomena is done by a set of some manually constructed lexical-syntactic text patterns such as "$\langle NOMINAL\_PHRASE \rangle$, $\langle NOMINAL\_PHRASE \rangle$,". In the case that some pattern (instantiated with the question and answer) matches the support text, then this Boolean feature is set to TRUE, in other case it is set to FALSE.

For instance, when the last text pattern is instantiated with the question *"What is the quinua?"* (only the question's target is used) and the candidate answer *"an American cereal of great nutritional value"*, the following text is obtained *"the quinua, an American cereal of great nutritional value,"*. This text matches the before mentioned support text and because that the feature that indicates the implicit relation detection is set to TRUE.

Term Overlap. This process calculates the term overlap between the support text and the hypothesis by a simple counting of the common words in the pair $(T,\ H)$. In order to avoid a high matching caused by functional terms (such as prepositions and determiners), it only considers the occurrence of content terms (nouns, verbs, adjectives and adverbs). This analysis allows generating the following six features: (1) the rate of noun overlap, (2) the rate of verb overlap, (3) the rate of adjective overlap, (4) the rate of adverb overlap, (5) the rate of date overlap, and (6) the rate of number overlap. See table 1 as example.

Structure Overlap. This process measures the surface structure overlap between the support text and the hypotheses. Similar to the term overlap process, it also only considers content words, but in addition, it takes advantage of the POS tags (see table 1 as example).

In order to compute this overlap we extract the longest common subsequence (LCS) between the support text and the hypotheses. In this case, it is necessary to compute the LCS from $(T,\ H)$ as well as from $(T,\ H')$. Nevertheless, only the longest subsequence is used. This way we generate the following feature from this analysis: the normalized size of the LCS between $(T,\ H)$ or $(T,\ H')$. That is, size of the LCS divided by the size of the longest subsequence in *H*.

4.3 Entailment Recognition

This final process generates the answer validation decision by means of a supervised learning approach, in particular, by a Support Vector Machine Classifier.

This classifier decides whether to accept or reject the candidate answer based on the ten previously described features along with the following two additional ones: the question category (i.e., factoid or definition) and the question interrogative particle (i.e., who, where, when, etc.).

An evaluation of the proposed features during the development phase – using the information gain algorithm – shows us that the question category and the question interrogative particle are between the five most discriminative features, while the nouns overlap and the LCS size are the most discriminative.

Table 1. Overlap analysis example (in the LCS the AJ, N and V are POS tags that indicates adjective, noun and verb, respectively)

Question	*"What is the quinua?"*
Answer	*"an American cereal of great nutritional value"*
Support text	*T="the quinua, an American cereal of great nutritional value,"*
Hypotheses	*H="an American cereal of great nutritional value is the quinua"*
	H'="the quinua is an American cereal of great nutritional value"
Term overlap	rate of nouns $= \frac{\lvert\{quinua, cereal, value\} \in H \cap T\rvert}{\lvert\{quinua, cereal, value\} \in H\rvert} = 1$
	rate of verbs $= \frac{\lvert\{\} \in H \cap T\rvert}{\lvert\{be\} \in H\rvert} = 0$
	rate of adjectives $= \frac{\lvert\{american, great, nutritional\} \in H \cap T\rvert}{\lvert\{american, great, nutritional\} \in H\rvert} = 1$
	rate of adverbs $= \frac{\lvert\{\} \in H \cap T\rvert}{\lvert\{\} \in H\rvert} = 1$
	rate of dates $= \frac{\lvert\{\} \in H \cap T\rvert}{\lvert\{\} \in H\rvert} = 1$
	rate of numbers $= \frac{\lvert\{\} \in H \cap T\rvert}{\lvert\{\} \in H\rvert} = 1$
LCS(H,H)	*american* AJ *cereal* N *great* AJ *nutritional* AJ *value* N *be* V *quinua* N
LCS(T,H)	*american* AJ *cereal* N *great* AJ *nutritional* AJ *value* N
LCS(T,H')	*quinua* N *american* AJ *cereal* N *great* AJ *nutritional* AJ *value* N
Structure overlap	normalized size $= \frac{\lvert LCS(T,H')\rvert}{\lvert LCS(H,H)\rvert} = 0.86$

5 Evaluation Results

In order to evaluate the proposed QA ensemble we used a set of 190 questions and the answers from 17 different QA systems[6]. In total, we considered 2286 candidate answers (with their corresponding support texts) for the evaluated questions. It is important to mention that this test set was employed at the first Spanish AVE (Answer Validation Exercise)[7], and that the system's responses were previously evaluated in the QA track at CLEF 2006 [2].

The main objective of our experiment was to demonstrate that our system ensemble could outperform each individual result. To evaluate the ensemble performance we used the accuracy measure. This measure is the most common evaluation metric for QA and indicates the percentage of correctly-answered questions[8] [13]. Table 2 shows the accuracy rates for each individual QA system. Internal columns show the system's accuracies for each type of question (F – factual, T – temporal restricted, D – definition).

Table 3 shows the accuracy results from different QA ensembles. The first three rows indicate some baseline results. In particular, the first row (ensemble 1) shows the results from an ideal ensemble, and the second and third lines (ensembles 2 and 3) shows the results achieved by two traditional ensembles. Finally, the last two rows (ensembles 4 and 5) indicate the results obtained by two variations of the proposed ensemble. The following paragraphs give a brief description and discussion on these ensembles.

[6] For the train phase we used the SPARTE corpus [12].

[7] We thanks the AVE organizers for provide us the answer-run id relations.

[8] An unanswered NIL question is considered as correctly answered.

Table 2. QA system's accuracies (to details about these systems refers to [14])

		% Right			
System	ID at CLEF 2006	F	T	D	ALL
1	alia061enes	17.59	12.50	40.48	21.58
2	alia061eses	37.04	22.50	38.10	34.21
3	aliv061eses	29.63	22.50	35.71	29.47
4	aliv062eses	21.30	22.50	28.57	23.16
5	aske061enes	6.48	2.50	7.14	5.79
6	aske061eses	16.67	15.0	11.90	15.26
7	aske061fres	12.96	5.0	7.14	10.0
8	inao061eses	47.22	35.0	83.33	**52.63**
9	lcc_061enes	20.37	25.0	14.29	20.0
10	mira062eses	10.19	12.50	23.81	13.68
11	mira061eses	21.30	15.0	16.67	18.95
12	pribe061eses	52.78	27.50	69.05	51.05
13	pribe061ptes	24.07	25.0	16.67	22.63
14	sinaiBruja06eses	16.67	17.50	33.33	20.53
15	upv_061eses	37.04	25.0	47.62	36.84
16	upv_062eses	27.78	25.0	40.48	30.0
17	vein061eses	32.41	25.0	83.33	42.11

Table 3. Ensemble's accuracies

		% Right			
Ensemble	Description	F	T	D	ALL
1	Ideal external ensemble	87.96	72.50	100.0	87.37
2	Based on systems confidence	52.78	35.0	83.33	55.79
3	Based on answers redundancy	51.85	27.50	52.38	46.84
4	Based on answer validation *(random)*	46.3	40.0	73.81	51.05
5	Based on answer validation *(ordered)*	51.85	42.50	85.71	**57.37**

Ensemble 1 is the ideal external ensemble. It indicates the maximum accuracy that can be reached by any external ensemble in the given test set. Its result is of great relevance since it confirms that current QA systems are complementary (it is possible to achieved 34% more accuracy than the best individual system).

Ensemble 2 is a confidence-based ensemble. Its output is the candidate answer extracted by the system having the greatest confidence value associated to the given type of question. Although this ensemble could outperform the best individual result by 3%, it has an important limitation: it does not take advantage of complementary systems for the same type of question, i.e., it does not contemplate that two or more systems can be good enough for answering an specific type of question.

Ensemble 3 is a redundancy-based ensemble. Its output is the most frequent candidate answer (or NIL if there is not a most frequent answer). This kind of ensemble allows taking into account the responses of all QA systems, and thus, their whole complementarity. However, it produced a very poor result, obtaining 6% less accuracy than the best individual result. A detailed analysis of this result showed us that even though only 19 questions were responded by just one system, the redundancies of the correct answers were very low (mainly because the same answer can be written in different ways). We also noticed that, given the low precision of most QA systems, in many cases incorrect answers had high redundancies. An additional problem emerged at the time of assigning the support text (for a frequent answer may exist several different support texts, in this case the problem is to select the most pertinent one).

Ensemble 4 is an ensemble based on answer validation (refer to section 3). Its result was disappointing; its overall accuracy was below the best individual result. We attribute this behavior to the fact that the answer validation module has a high recall (73%) but a very low precision (52%)[9]. Therefore, the strategy of selecting as final response the first validated answer is not adequate, since this answer has great probability (48%) of being erroneous. However, it is important to point out that there is also a great probability of capturing the correct response in one of the subsequent accepted answers.

Ensemble 5 is an extension of Ensemble 4. It introduces a simple modification that allows avoiding the problems caused by the low precision of the answer validation module. Different from Ensemble 4 that evaluates the answers in a random order, this new ensemble takes answers in a decreasing order based on the general confidence (accuracy) of their source QA system. The result achieved by this ensemble was very significant. It outperformed the best individual result by almost 5% and was better that all previous ensemble results.

6 Conclusions

In this paper we proposed an external QA ensemble based on answer validation. Like other external ensembles, it does not rely on internal system's features.

[9] That indicates that this module detects most correct answers but also validates several incorrect answers.

Nevertheless, it distinguishes from these ensembles in that: (*i*) it does not depend on the answer's redundancies, (*ii*) it is not obligated to always select one candidate answer, and (*iii*) it not only allows returning correct answers but also supported ones.

The evaluation results demonstrated the appropriateness of our proposal. Although the current validation module is still very imprecise, our QA ensemble (using an ordered set of candidate answers) could outperform the best individual result as well as the results from traditional ensemble approaches.

It is important to notice that an increment on the answer validation precision will directly impact on the ensemble accuracy. Based on this observation, our future work will be focused on improving this module. In particular, we plan to include other features for the entailment recognition such as the edit distance between the syntactic trees of T and H, and to calculate an accepted confidence value based on the most discriminative features used for the textual entailment recognition.

Acknowledgments

This work was done under partial support of CONACYT (project grant 43990 and scholarship 171610). We also like to thanks to the CLEF organizing committee as well as to the EFE agency for the resources provided.

References

1. Vallin, A., Magnini, B., Giampiccolo, D., Aunimo, L., Ayache, C., Osenova, P., Peñas, A., de Rijke, M., Sacaleanu, B., Santos, D., Sutcliffe, R.F.E.: Overview of the clef 2005 multilingual question answering track. In Peters, C., Gey, F.C., Gonzalo, J., Müller, H., Jones, G.J.F., Kluck, M., Magnini, B., de Rijke, M., eds.: CLEF. Volume 4022 of Lecture Notes in Computer Science., Springer (2005) 307–331
2. Peñas, A., Rodrigo, Á., Sama, V., Verdejo, F.: Overview of the answer validation exercise 2006. [14] 257–264
3. Magnini, B., Negri, M., Prevete, R., Tanev, H.: Is it the right answer? exploiting web redundancy for answer validation. In: ACL. (2002) 425–432
4. Dietterich, T.G.: Machine-learning research: Four current directions. The AI Magazine **18**(4) (1998) 97–136
5. Pizzato, L.A.S., Molla-Aliod, D.: Extracting exact answers using a meta question answering system
6. Chu-Carroll, J., Czuba, K., Prager, J.M., Ittycheriah, A.: In question answering, two heads are better than one. In: HLT-NAACL. (2003)
7. Rotaru, M., Litman, D.J.: Improving question answering for reading comprehension tests by combining multiple systems. In: In Proceedings of the American Association for Artificial Intelligence (AAAI) 2005 Workshop on Question Answering in Restricted Domains, Pittsburgh, PA. (2005)
8. Jijkoun, V., de Rijke, M.: Answer selection in a multi-stream open domain question answering system. In McDonald, S., Tait, J., eds.: ECIR. Volume 2997 of Lecture Notes in Computer Science., Springer (2004) 99–111

9. Aceves-Pérez, R.M., y Gómez, M.M., Pineda, L.V.: Graph-based answer fusion in multilingual question answering. In Matousek, V., Mautner, P., eds.: TSD. Volume 4629 of Lecture Notes in Computer Science., Springer (2007) 621–629
10. Dagan, I., Magnini, B., Glickman, O.: The pascal recognising textual entailment challenge. In: Proceedings of Pascal Challenge Workshop on Recognizing Textual Entailment, Southampton, UK (April 2005) 1–8
11. Carreras, X., Chao, I., Padró, L., Padró, M.: Freeling: An open-source suite of language analyzers. In: Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC'04), Lisbon, Portugal (2004)
12. Peñas, A., Rodrigo, Á., Verdejo, F.: SPARTE, a test suite for recognising textual entailment in spanish. In Gelbukh, A.F., ed.: CICLing. Volume 3878 of Lecture Notes in Computer Science., Springer (2006) 275–286
13. Magnini, B., Giampiccolo, D., Forner, P., Ayache, C., Jijkoun, V., Osenova, P., Peñas, A., Rocha, P., Sacaleanu, B., Sutcliffe, R.F.E.: Overview of the CLEF 2006 multilingual question answering track. [14] 223–256
14. Peters, C., Clough, P., Gey, F.C., Karlgren, J., Magnini, B., Oard, D.W., de Rijke, M., Stempfhuber, M., eds.: 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. In Peters, C., Clough, P., Gey, F.C., Karlgren, J., Magnini, B., Oard, D.W., de Rijke, M., Stempfhuber, M., eds.: CLEF. Volume 4730 of Lecture Notes in Computer Science., Springer (2007)