Knowledge-Based Systems 137 (2017) 83–93

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

A study about the future evaluation of Question-Answering systems

Alvaro Rodrigo*, Anselmo Peñas

NLP & IR Group at UNED, Juan del Rosal 16, Madrid, Spain

ARTICLE INFO

Article history:
Received 22 December 2016
Revised 29 July 2017
Accepted 8 September 2017
Available online 9 September 2017

Keywords:
Question Answering
Evaluation campaigns
Validation
Textual inference

ABSTRACT

Evaluation campaigns of Question Answering (QA) systems have contributed to the development of such technologies. These campaigns have promoted some changes oriented to overcome results. However, at this period we see how systems have reached an upper bound, as well as systems are still far away from answering complex questions. In this paper, we overview the main QA evaluations over free text, paying special attention to the changes encouraged at such campaigns. We observe that systems still return a high proportion of incorrect answers and that the changes are almost not included in traditional approaches. Moreover, we analyze QA collections in order to obtain better insights about the main challenges for current QA systems. We detect that QA systems find very difficult to deal with different rewordings in questions and documents, as well as to infer information that is not explicitly mentioned in texts. Based on those observations, we recommend a set of directions for future evaluations, suggesting the application of textual inference and knowledge bases as a way for improving results.

1. Introduction

Question Answering (QA) has attracted widespread interest given the vast amount of digital documents and Web pages available in the last decades. QA systems focus on finding exact answers to questions formulated in Natural Language. The proposal of evaluation campaigns for these systems, in addition to the development of Information Retrieval (IR) technologies, contributed to obtain big advances in results. However, QA systems usually showed an upper-bound performance close to a 60% of questions correctly answered [1]. Consequently, users did not trust these systems.

Some researchers introduced new evaluation campaigns aimed at promoting new QA systems able to overcome such upper bound. These campaigns focused on improving the self-confidence of QA systems as a way of reducing incorrect answers. For example, ResPubliQA introduced *c@1* as the main evaluation measure [2]. *c@1* rewards QA systems that reduce incorrect answers while maintaining correct ones. Additionally, some researchers suggested simple and individual tasks with the goal of helping to develop better QA systems [3]. On the other hand, some systems reached promising results in isolated tasks out of the scope of traditional campaigns, as for example Watson, the IBM's QA system which defeated humans at the Jeopardy TV show [4]. Despite some of these individual efforts, there has not been a clear improvement in systems' results evaluated under the same conditions. In fact, nobody, to our knowledge, has studied evaluation campaigns in order to find the main difficulties for systems.

In this paper, we analyze the last evaluation campaigns of QA systems and detect the main barriers for current technologies. On the basis of these issues, we recommend a set of incremental evaluations addressed to foster new QA systems able to improve current results.

The main contributions of this paper are:

- Describe the evolution of evaluations focused on promoting better QA results.
- Remark the motivations and main lessons learned from each evaluation.
- Analyze an evaluation collection and identify the major problems for current technologies.
- Collect open research directions that should be explored in order to advance in the QA field.

The rest of this paper is organized as follows: We firstly described in Section 2 the methodology and main measures employed to evaluate QA systems. Then, Section 3 reports the first proposals of QA over free-text. Section 4 shows some movements of the Natural Language Processing community towards the development of Answer Validation (AV) technologies aimed at improving QA results. Section 5 presents a proposal for encouraging a better self-confidence of QA systems by leaving some questions unanswered instead of returning incorrect answers. In Section 6, we focus on Answer Validation in the QA context using Reading Comprehension tests, while Section 7 shows the importance of Back-

* Corresponding author.
E-mail addresses: alvarory@lsi.uned.es (A. Rodrigo), anselmo@lsi.uned.es (A. Peñas).

http://dx.doi.org/10.1016/j.knosys.2017.09.015
0950-7051/

Table 1
Evaluation measures according to the number of responses and different uses of the systems self-confidence.

	Number of answers per question		
	1	> 1 permitted but only 1 is needed	> 1 are needed
Without ranking	Accuracy [6]		Precision, Recall, F [6]
Ranking by question		MRR [7]	
Ranking for all questions	CWS [8]		
Cardinal confidence self-score	K1 [9]	K [9]	K [9]
Unanswered different from incorrect	c@1 [10]		

ground Knowledge in QA. Section 8 describes the analysis of a collection in order to discover the main challenges for QA systems. Then, in Section 9 we offer a discussion based on the main observations obtained across this work, collecting future directions for improving QA technologies. In Section 10, we give a short view of related QA approaches and evaluations. Finally, some conclusions are given in Section 11.

2. Evaluation methodology

Evaluations for free-text QA are based, as IR evaluations, on the Cranfield paradigm [5]. These evaluations make available a document collection and a set of questions. Then, participant systems return answers extracted from those documents. These answers are assessed in order to judge their correctness with respect to the given questions. Finally, depending on the correctness of these answers, a final score for each system is given.

The method to obtain the final score depends on the evaluation measure selected. Each measure evaluates a different set of features. Hence, researchers must select the measure depending on the objectives of the evaluation. In the next Subsections, we describe the main evaluation measures used in QA. Furthermore, we show in Table 1 the main features of each measure described in this paper.

2.1. Measures focused on correct answers

The main QA measure has been *accuracy*, which represents the proportion of questions correctly answered. The formulation of *accuracy* is shown in Eq. (1), where n_{ac} is the number of questions correctly answered and n is the number of questions. This definition[1] of *accuracy* restricts the number of answers to one per question. Although *accuracy* is a simple and intuitive measure,it only acknowledges correct answers and does not consider the amount of incorrect answers.

$$accuracy = \frac{n_{ac}}{n} \tag{1}$$

On the other hand, *Mean Reciprocal Rank* (*MRR*) was used when several answers per question are given in a ranking list [7]. According to *MRR*, each question receives a score that is the reciprocal of the rank at which the first correct answer is given, or 0 if no correct answer is returned. This score represents the Reciprocal Rank (RR) for each question. The final score is the mean over the Reciprocal Ranks of all the questions. Although *MRR* acknowledges systems able to place correct answers at high positions in the raking, it stimulates also the risk of giving wrong answers instead of not responding. This is because a wrong answer receives the same value that an empty answer. Since a random answer might be correct, and therefore, that answer would add some value to the final score, it is better to give a random answer than to give an empty answer.

[1] Some QA researchers used another definition of *accuracy* that allows several answers per question, but this definition was not used in common evaluations.

2.2. Measures based on self-confidence scores

In order to overcome limitations of the aforementioned measures, some evaluation campaigns suggested measures based on *self-confidence scores*. TREC proposed *Confidence Weighted Score* (*CWS*) [8], which is based on *Average Precision*. *CWS* requires systems to return only one answer per question and rank all the answers according to system's self-confidence. Then, *CWS* rewards systems returning correct answers at top positions in the ranking. *CWS* is defined in Eq. (2), where n is the number of questions and $C(i)$, defined in Eq. (3), is the number of correct answers up to the position i in the ranking. In Eq. (3), $I(j)$ is a function that returns 1 if answer j is correct and 0 if it is incorrect.

$$CWS = \frac{1}{n}\sum_{i=1}^{n}\frac{C(i)}{i} \tag{2}$$

$$C(i) = \sum_{j=1}^{i} I(j) \tag{3}$$

CWS gives more value to some questions over others because questions with answers at top positions contribute more to the final score. This is why *CWS* was discussed as a measure for evaluating QA systems and discarded in the following campaigns of TREC, which recovered *accuracy*. However, *CWS* has been employed for evaluating individual systems [11].

On the other hand, CLEF proposed the use of the *Pearson's correlation coefficient* and two new measures: *K* and *K1* [9]. *K* and *K1* are based on a utility function that returns -1 if the answer is incorrect and 1 if it is correct. Both measures weight this value with the confidence score given by the system.

The use of the *Pearson's correlation coefficient, K* and *K1* did not success because it is difficult to understand the final score. In fact, only CLEF used these measures [12]. A positive value does not indicate more correct answers than incorrect ones, but that the sum of scores from correct answers is higher than the sum of scores from incorrect answers.

2.3. Acknowledging non-response

QA systems with a good self-confidence should be able to detect and reduce incorrect answers while keeping, or even rising, correct answers. A method for evaluating the self-confidence of QA systems is to give value to the fact of *not answering* instead of giving incorrect answers. This method rewards systems that reduce incorrect answers and keep correct ones. In order to do it, systems must leave some questions unanswered. More in detail, if a system prefers not to answer a question because the system is unsure about finding correct answers, the system should receive a better score than in the case of returning an incorrect answer. This is a behavior of great importance in scenarios where an incorrect answer may have associated a high cost or risk, as for instance in medical diagnosis.

Correctness at one (*c@1*) was proposed according to the observations above [10]. In this Subsection, we expose the nature and main features of *c@1*.

c@1 was defined for QA scenarios with the following features:

- There are several questions.

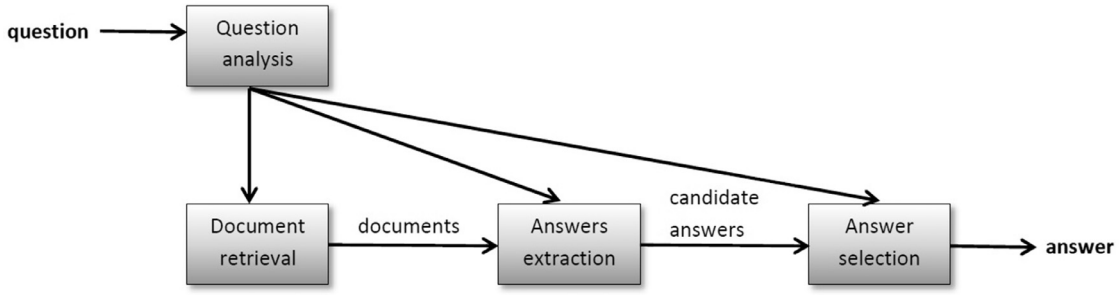

Fig. 1. Schema of the classical architecture of Question-Answering systems.

Table 2
Contingency table for the detailed scenario.

	Correct (C)	Incorrect (¬C)
Answered (A)	n_{ac}	n_{aw}
Unanswered (¬A)		n_u

- Only one relevant answer per question is requested and all the questions have at least a correct answer.
- A system can decide not to answer a question if the system is not confident about finding a correct answer.

Since every question has a correct answer in the proposed scenario, the non-response option is not correct but it is not incorrect either. This behavior is represented in contingency Table 2, where:

- n_{ac} is the number of questions correctly answered.
- n_{aw} is the number of questions incorrectly answered.
- n_u is the number of unanswered questions.
- n is the number of questions ($n = n_{ac} + n_{aw} + n_u$).

According to Table 2, *c@1* is formally represented in Eq. (4):

$$c@1 = \frac{n_{ac}}{n} + \frac{n_{ac}}{n}\frac{n_u}{n} = \frac{1}{n}\left(n_{ac} + \frac{n_{ac}}{n}n_u\right) \quad (4)$$

The most important features of *c@1* are:

1. A system that answers all the questions would receive a score equal to the one obtained with *accuracy* because n_u=0 and therefore, $c@1$=n_{ac}/n.
2. Unanswered questions add value to *c@1* as if they were answered with the *accuracy* already shown.
3. A system that do not return any answer would receive a score equal to 0 due to the fact that n_{ac}=0 in both addends.

c@1 rewards systems that leave questions unanswered in order to reduce incorrect answers and keep correct answers. The more correct answers given, the most confident is the system about their answers and then, unanswered questions receive more value. Thus, *c@1* acknowledges systems with a high confidence in their answers.

c@1 has been also used for evaluating plagiarism detection [13].

3. From knowledge bases to Question-Answering based on information retrieval

In this section, we describe the first efforts in the evolution of QA systems, with a special focus on evaluation campaigns over free-text.

3.1. First Question-Answering systems

The first QA systems were developed in the 1960s, and they were used as Natural Language interfaces to databases and Knowledge Bases (KB) over restricted domains. Two of the most famous proposals were BASEBALL [14], which gave responses about the US baseball league, and LUNAR [15], which answered questions about the geological analysis of the moon.

More complex QA systems were developed in the 1970s and 1980s. These systems took advantage of advances in computational linguistics. Some examples of these technologies were UNIX Consultant (UC) [16] and LILOG [17]. While UC answered questions about the UNIX operating system by adapting output strings to user profiles, LILOG answered questions about tourism in Germany's cities. However, at that time, the amount of knowledge available in structure resources was limited and domain restricted.

3.2. Question Answering based on information retrieval

In the 90's, the huge amount of electronic documents attracted a general interest on technologies able to extract information from texts. IR systems, as for example Web search engines, are one of the most important examples of such technologies. IR systems find text documents which satisfy information needs from users [18]. Advances in IR fostered the research on QA systems that work over non-restricted domain free-texts. While IR systems return a list of documents where users must look up answers, QA technologies return a short text containing the answer [19]. During this period, QA systems developed a strong dependency on the IR component, which determined QA architectures and permitted big advances during a long period of time.

Typical QA architectures over free text are based on *pipeline* processing. The common components of these architectures are: *question analysis, retrieval* of candidate documents, *answer extraction* and *answer selection* [20–22]. We show in Fig. 1 a schema representing the main components of classical architectures. These architectures filter the texts returned from IR engines with the purpose of finding correct answers at the end of the pipeline.

The series of QA evaluations at the Text REtrieval Conference (TREC)[2], whose first edition took place in 1999, represented an important event in QA research. Similar evaluations were proposed at the Cross Language Evaluation Forum (CLEF)[3] and at the NII Test Collection for IR Systems (NTCIR)[4], including multilingual and cross-lingual tasks. These evaluations mainly proposed a set of factoid questions, as for example "*Who is the president of X?*" or "*When was Y born?*". Then, participant systems returned answers extracted from documents. These evaluations focused on *acknowledging only correct answers* without penalizing incorrect ones, which promoted systems returning answers no matter these answers could be incorrect.

Nowadays, some of these evaluation campaigns have focused on other issues such as evaluating real time systems [23] or QA over restricted domains [24].

[2] http://trec.nist.gov/.
[3] http://www.clef-initiative.eu/.
[4] http://research.nii.ac.jp/ntcir.

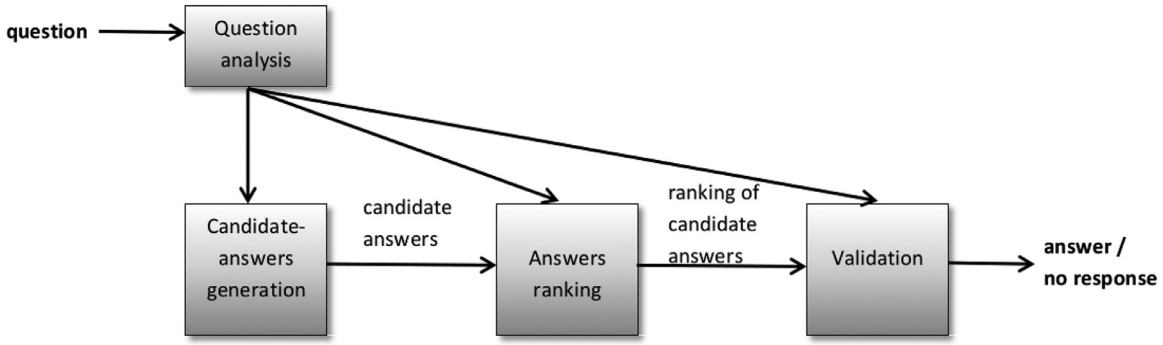

Fig. 2. Schema of the new architectures based on answer generation and validation.

4. From retrieval to validation

In this Section, we describe the changes proposed in evaluations with the objective of encouraging changes in QA. These changes tried to convince researchers to use the new QA architectures described in this Section instead of the classical QA architectures described above.

QA architectures based on *pipeline* processing, as it was described in Section 3.2 and shown in Fig. 1, are highly affected by error propagation. For instance, a QA system using retrieval and answer extraction modules with an individual performance of 80% would have an upper bound performance of 64% due to dependencies among modules. Besides, the improvement of a single component does not guarantee better results [25]. Moreover, QA systems based on IR engines require that the wording of texts containing an answer must be similar to the wording used in the question.

In order to break these limitations, some researchers at the Natural Language Processing (NLP) community suggested to include validation in QA. For example, the organizers of the First Recognizing Textual Entailment (RTE) Challenge recommended validating QA responses using RTE systems, which cast the inference between texts as a textual entailment problem [26]. The inclusion of validation modules enabled new *architectures based on (1) over-generation of candidate answers* as hypotheses, and *(2) validation*. We show in Fig. 2 a schema representing this idea of new architectures.

The objective of this tendency is to promote a new kind of systems. These new systems would be able to perform an intelligent search across the space of candidate answers. It is important to remark that this search would not be only restricted to results from an IR engine. After generating all possible answers, these new architectures would rank the candidate answers. Finally, a validation module takes the final decision. That is, the validation module decides if the system return an answer. This decision depends on different aspects, working scenarios, etc. For example, we can imagine a scenario where an incorrect answer represents a high risk, as it happens with a false negative in medical diagnosis. In such scenario, a system should return an answer only if the system has a very high confidence on the answer's correctness. *IBM's Watson* followed this idea in its participation at Jeopardy [4]. Watson had a first step focused on over-generating all possible hypotheses with the objective of ensuring correct answers. At the second step, Watson ranked candidates and applied validation in order to decide if there was a final answer.

Although validation represented a key component in new architectures, it was a new area of research for the QA community. In next Subsections, we define the validation task and describe the first proposals of validation systems. Finally, we describe a validation task that fostered the development of validation technologies and their inclusion in QA systems.

4.1. Definition of answer validation

An Answer Validation module receives a $<$Question,Answer$>$ pair and returns a value indicating if the Answer is correct or not [27]. AV can be seen as a *binary classification* task where answers must be classified as correct or incorrect. The main objective of AV technologies is to reduce the amount of incorrect answers given by QA systems, which produces an improvement in QA performance and user's confidence [28].

AV modules can perform two different subtasks:

- *Validation*: an AV module in this subtask discards incorrect answers from a set of candidates. This subtask is employed in scenarios where QA systems can return multiple answers per question. Thus, the AV module is used as a filter. In this setting, we say that an AV system *validates* an answer if the AV system considers the answer as correct. We say that the AV module *rejects* an answer if the AV system considers the answer as incorrect. Validated answers are returned, whereas rejected answers are discarded.
- *Selection*: an AV module in this subtask returns no more than one final answer among a set of candidates. For this subtask, an AV module firstly discards incorrect answers and then, selects the most promising answer. This subtask is used in scenarios where QA system must return no more than an answer per question.

4.2. Approaches to validation

We review in this Section the first proposals of validation systems. According to their characteristics, we divide these proposals into three main approaches:

1. Systems counting *redundancies*.
2. Systems measuring the *similarity* of the answer with a supporting snippet.
3. Systems performing a deeper *textual analysis* between questions and answers.

Next subsections describe in more detail these three approaches.

4.2.1. Approaches based on redundancy

The first validation approaches assumed that the candidate answer with more repetitions was the more likely to be correct. For example, [29] applied a *voting* scheme among candidate answers.

Other systems relied on the Web as a huge resource of human knowledge that contains different reformulations about the same fact. These systems extracted keywords from questions and answers and combined them into web-search queries [27]. These technologies worked under the assumption that *the more documents extracted with such keywords, the more likely to be correct the*

answer [30]. However, this assumption is not always true and all these approaches were not able to approximate the deeper semantic phenomena required to determine answerhood [28].

4.2.2. Approaches based on textual similarity

Most of the approaches described in this Subsection were developed under the AV evaluation campaigns described in Section 4.3. These methods measured the similarity between (1) a combination of the answer with the question and; (2) a text snippet supporting answer's correctness. These approaches consist of *two main components*:

- A module measuring *similarities* at different linguistic levels.
- A module taking the *final validation decision* according to a set of features extracted from the previous component.

Similarity modules worked at lexical, syntactic and semantic levels. While most of these modules only measured lexical overlapping among words and/or n-grams [31], some modules considered *syntactic* features such as overlapping in dependency trees. Hardly any systems worked at *semantic* level, with techniques such as semantic role labeling and semantic parsing [32].

The final decision about validation relied on Machine-Learning techniques. Although the most common classifiers were *Support Vector Machines* (SVM) and *Decision Trees*, no clear evidence about a best performance of them over other classifiers was found.

4.2.3. Approaches based on text analysis

These methods analyzed connections between questions and answers. For instance, Harabagiu and Maiorano [33] validated an answer only if it was possible to find an explanation for the answer using *abductive inference*. The proposed abduction process was based on the interpretation of appositions and lexical-syntactic patterns for hyponymy and troponymy. Other methods transformed questions and answers into logical forms and applied a *theorem prover* that included world knowledge extracted from WordNet[5]. If a proof of the answer was found, then the answer was considered correct [34]. Although these systems achieved some of the best results in this task [35], they depended on vast amounts of background knowledge and they had a high computational cost associated to obtain logical proofs [36].

4.3. Evaluating answer validation

In this Section, we describe the Answer Validation Exercises (AVE) at CLEF. AVE was proposed in 2006 as an evaluation task to advance the state of the art of validation technologies [37]. The objective of AVE was to produce a change in QA architectures giving more responsibility to validation. For this purpose, AVE assumed a previous step of candidate answers over-generation, leaving the hard work to the validation step.

The starting point of AVE was the *reformulation* of Answer Validation as an RTE problem. AVE worked under the assumption that RTE hypotheses can be automatically generated by combining questions with candidate answers. Thus, AVE proposed the inclusion of Machine Learning techniques into QA given the binary-classification nature of the validation task. In addition, AVE suggested the application of deep analysis techniques to detect correct answers not explicitly stated in texts.

The methodology of AVE took into account the *unbalanced nature of collections* in the *validation setting*, where correct and incorrect answers might be found in a different proportion. For example, Breck et al. [38] observed a 78% of incorrect answers in the TREC-8 QA output, while AVE organizers reported an 86% of incorrect answers in the AVE 2008 English collection, which was generated from the real output of QA systems [39]. This unbalanced nature of collections is also a problem for Machine Learning methods, which have to adapt their configuration to this kind of collections [40].

In the validation setting, AVE applied the harmonic mean (*F-measure*) of *precision* and *recall*. This decision was motivated by the fact that *accuracy* assumes that the classes in the target environment are relatively balanced.

In the selection setting at AVE, AV systems had to select an answer per question from a pool of candidates generated from real QA systems. Then, AVE organizers compared the resulting behavior with the performance of real QA systems. This permitted to study the impact of AV in QA results. For this purpose, evaluation in the *selection setting* was based on traditional QA *accuracy*. This was because *accuracy* measures the proportion of questions that have received a correct answer, and therefore, it is not affected by unbalanced classes in the collection.

One of the main contributions of AVE was to suggest how the potential performance of QA systems including AV modules could be estimated. This estimation assumed that AV modules would request new candidates if no correct answer was found. Results of participants at AVE showed how AV could help to improve QA performance, encouraging the inclusion of validation in some real QA systems at CLEF [41,42].

Newer studies have proposed new methods for analysing the impact of validation on QA results [43]. These studies reused AVE collections.

In this section, we have shown that validation technologies can discard incorrect answers from QA systems. This capacity would allow QA systems to improve results. Thus, the next step was to foster the inclusion of validation technologies in QA.

5. A first attempt to introduce validation in Question Answering

After developing individual validation modules, it was time to promote its inclusion in real QA systems. This was the main motivation for proposing the ResPubliQA campaing at CLEF in 2009 [2]. In this Section, we describe the objectives and main lessons learned at ResPubliQA. We focus on: (1) how ResPubliQA tried to promote the inclusion of validation; (2) if participants systems finally included validation and; (3) the impact of validation in results.

ResPubliQA tried to foster the inclusion of validation technologies by rewarding QA systems with a correct self-confidence. For this purpose, ResPubliQA included the option of leaving unanswered some questions. Then, given two systems that returned the same proportion of correct answers, the system with less incorrect answers, by leaving unanswered some questions, scored better. This is why *c@1* was the main evaluation measure at ResPubliQA.

Participants at ResPubliQA returned no more than one response per question. Each question had to receive one of the following kinds of response:

- A paragraph with a candidate answer.
- The string *NOA* indicating that the system preferred not to answer the question.

Systems preferring to leave a question unanswered could optionally submit a candidate answer. In such cases, *accuracy* was used as a secondary measure. *Accuracy* considered the hypothetical answers given to unanswered questions, which enabled organizers to evaluate the validation performance.

[5] https://wordnet.princeton.edu/.

The best performing systems at ResPubliQA obtained scores of 0.6 of both *c@1* and *accuracy*. These results were similar to the best scores obtained in previous QA campaigns at CLEF. However, the perfect combination of systems showed that a 90% of questions received a correct answer by at least one system. This result suggested that the task was feasible, but systems still had room for improvement.

Organizers offered a baseline based on a pure IR approach [44], with two objectives:

1. To test how well a pure IR system can perform at this task.
2. To compare the performance of more sophisticated QA technologies against a simpler IR approach.

This baseline was compared with some participant systems to show the main differences between *c@1* and *accuracy*. For instance, there was an English system which answered correctly 20 questions less than the baseline over a set of 500 questions. However, this system was able to reduce the number of incorrect answers in a significant way, returning 32 incorrect answers less than the baseline. *c@1* rewarded this behavior, produced a swap in rankings with respect to *accuracy* and gave a better ranking to the system. This observation manifested some of the good features of *c@1*.

Although the main objective of ResPubliQA was to promote a higher inclusion of validation technologies in QA, the consideration of unanswered questions and the application of *c@1* were not enough. Systems continued relying on pipeline architectures based on IR engines. Moreover, most of participants did not apply validation, and participants using validation were not able to improve results significantly. This was due to the fact that validation modules were not able to overcome limitations produced by pipeline architectures based on IR, where it is difficult to extract answers with a wording different to that in the question.

As with previous evaluations, datasets of ResPubliQA are available. These datasets have been used for evaluation purposes [45,46]. Nevertheless, researchers have used these collections more for evaluating passage retrieval than for evaluating validation.

In this Section, we have reported the first attempt to include validation in QA. However, this attempt failed to achieve the change expected. Furthermore, results were under the 60% upper bound. In fact, several systems performed worse than the pure IR baseline. Organizers recommended that changes in QA architectures required a previous development of answer validation/selection technologies in the QA context without paying so much attention to the IR step, which narrows the overall performance.

6. Leaving aside information retrieval and giving more importance to validation

In this Section, we describe the development of an evaluation campaign focused on promoting a higher incorporation of validation in QA systems. For this purpose, the described evaluation proposed to leave aside IR components. Thus, this evaluation tried to encourage a major inclusion of new architectures (described in Section 4 and shown in Fig. 2). This new effort worked on the main lessons learned from ResPubliQA.

A suitable scenario for evaluating validation in QA is represented by *Multiple choice* tests [47]. This is because multiple choice questions offer a set of candidate answers, where a system must detect correct candidates. This format is usually employed in Reading Comprehension (RC) tests focused on assessing the understanding of documents [48], as for instance when learning foreign languages. This observation motivated to move from ResPubliQA into a campaign based on RC tests, the Question Answering for Machine Reading Evaluation (QA4MRE), whose first edition was held in 2011 [49].

QA4MRE proposed to split the QA problem into generation and validation, the two main components of new architectures. While the hard work was left to the validation component, generation was simulated by multiple choice questions. For this purpose, QA4MRE proposed a set of multiple-choice RC tests.

Each participant system had to select an answer per question or let it unanswered, as in ResPubliQA. Since candidate answers were provided and there was always a correct answer, a question could be left unanswered if a system decided it was unable to find a correct answer. This represented an opportunity for applying validation technologies and reducing the amount of incorrect answers. This is why *c@1* was chosen as the main evaluation measure.

Most of participants decided to rank candidates and select the most promising answer. Systems utilized approaches based on counting similarities at surface levels, which is not enough for deciding about answers' correctness. We think that these inappropriate approaches contributed to the low performance. In fact, average results showed that participant systems returned more incorrect answers than correct answers.

7. The role of background knowledge in Question Answering

In this Section, we focus on the importance of considering background knowledge for answering questions.

The knowledge required for a complete understanding of a document is not always explicitly contained in it [50]. This is something typical in human language. We voluntarily omit some pieces of information in our messages because the receiver does not need them for a complete understanding. The receiver uses context and their own background knowledge to obtain an interpretation of the message [51]. A clear example is represented by the use of some noun compounds, as for example "*German people*" instead of "*people who were born in Germany*". The omission of that information supposes a big problem for automatic systems working with the meaning of a message or a whole document. These systems must deal with the problem of *completing knowledge gaps* in the document, which is a task that has not been solved yet.

In the context of RC tests, while the correct answer is usually present in the test document (albeit in a different form that the expected), it might be required additional information about the topic to know what to search for and how to do it [50]. In general, more prior background knowledge should make understanding and QA easier. This has been shown in the NLP community with the use of resources such as *Wordnets* [52,53], *FrameNets* [54,55], *paraphrase lists* [56], *knowledge bases* [57], etc; which offer different kinds of prior knowledge.

Given this relevant role of knowledge in RC tests, the organizers of QA4MRE provided a large collection of related documents. The purpose was to encourage the acquisition of background knowledge from those documents. Background knowledge in the context of QA4MRE was defined in terms of the relation between questions, answers and the background collection provided [49].

Results suggested that the inclusion of external sources and their exploitation were not widely adopted. In fact, participant systems did not report significant improvements employing the background collection and, other external resources seemed more promising than the provided collection. Maybe there were difficulties for extracting useful knowledge from the collection or about how to apply that knowledge. Hence, it is necessary to think about the use of such collections for acquiring knowledge in the context of QA.

8. Detecting the main problems for better Question Answering

In this Section, we concentrate on a deep study of a QA collection. Our objective is to detect the main difficulties for current technologies.

The observation of bad results over RC tests led us to analyze an RC collection with the purpose of discovering the main difficulties for systems. We performed this study over the test set of the *Entrance Exams* pilot task celebrated at the QA4MRE 2013 [58]. This task used real tests from English exams to access the Japanese University. The format of these tests is quite similar to those at the QA4MRE main task: a document and a set of questions about it, where questions are given in a multiple-choice format with four candidates per question and only one correct answer.

The main objective of *Entrance Exams* was to evaluate systems in the same conditions that humans do with RC tests. This is why this task could be seen as a Turing Test [59] where systems must emulate human intelligence answering questions about the understanding of a document. We preferred Entrance Exams tests instead of QA4MRE collections because Entrance Exams used real tests for humans. Moreover, this collection is similar to other posterior collections based on standardized tests to measure artificial intelligence [50].

Firstly, we observe that Entrance Exams tests are more challenging, even for humans, than the extraction of correct answers from texts (the task described in Section 3.2). In fact, results at *Entrance Exams* were quite low, with an average *c@1* score of 0.25 and a maximum of 0.42[6], which are similar to the average results at the QA4MRE main task. These tests require a *deep understanding* in order to avoid confusions between similar but not semantically equivalent texts.

We describe in next subsections the main difficulties observed in our study. We focus on those questions where systems showed a low performance. Then, we suggest in Section 9 some features for evaluations aimed at overcoming these difficulties.

8.1. Complex questions

Participant systems at Entrance Exams obtained low scores over complex questions with respect to easier questions. Complex questions refer to different pieces of information that must be gathered from several sources, as for example documents and knowledge bases [60]. Furthermore, these questions require the application of several inference steps, where each step might collect information demanded by the following steps. For instance, the question "*Who is the son of the President of the first developed country?*" requires systems to answer the following questions:

1. *What is the most developed country?*, whose answer is *X*.
2. *Who is the President of X?*, whose answer is Y.
3. *Who is Y's son?*

Both processes of searching and combining information make complex questions more challenging than atomic factoid questions as for example "*Who is the president of Germany?*", whose answer can be directly extracted from the text. Although the QA community made some proposals focused on answering complex questions several years ago [61,62], system's performance over these questions remains quite similar. This is why some researchers proposed to firstly develop QA systems able to answer more simple questions [3].

8.2. Textual inference

We have also detected poor results over questions where correct candidates appear with different wording in documents. This rewording is common in tests oriented to assess document understanding. Moreover, these questions may ask about information that appears implicitly, but not explicitly, in texts. For example, the Entrance Exams collection contains the candidate "*nature is still far more powerful than our technology*" that refers to "*human technology is unable to compete with natural forces*" in the test document. This example is easy for humans, but not for automatic systems.

This kind of questions requires textual inference and background knowledge to detect answers semantically equivalent to the text. However, most of QA technologies are based on finding answers that are explicit in the text [63].

8.3. Developing strategies for answering

We have also studied human behavior over the Entrance Exams collection, because we think that this behavior might be useful for building better QA systems. This is because humans usually develop strategies for answering difficult questions and we think some of these strategies could be included in automatic systems. In fact, some QA researchers have successfully imitated human strategies in their systems [61].

In our study, we have realized humans successfully employ rejection to solve RC tests. Rejection enables a strategy based on discarding incorrect answers and finally, selecting the only candidate that could not be rejected, or a candidate among the most promising ones. In fact, this is the idea of answer validation (as it was described in Section 4).

We observed that the rejection strategy permits humans to have better chances answering some questions where systems achieved a low performance. However, we think current QA systems are applying naive validation strategies. Hence, it is worth developing more sophisticated validation strategies to overcome current results.

9. Discussion and open research challenges

In this Section, we collect some of the open research directions that are worth exploring in order to advance the state of the art in this field. These directions are based on the observations obtained from previous Sections, with a special focus on those at the previous Section.

As it has been observed during this work, evaluations tried to drive a change towards *new QA architectures* encouraging:

1. Systems *generating hypotheses* as candidate answers;
2. Systems able to decide whether a candidate answer is *correct, incorrect* or the system *cannot decide* about its correctness;
3. Systems able to decide that there is *no correct answer* among the set of candidates;
4. Systems able to reduce incorrect answers by leaving some questions unanswered.

The QA community pointed out the importance of some of these features [64], especially after IBM's Watson participation at the Jeopardy Challenge. As it was pointed out by the IBM team, this challenge required a great ability to "*know what you know*" [4].

From an evaluation perspective, the change to new QA architectures has been addressed with methodological proposals, as for instance the Answer Validation Exercise (described in Section 4.3) and the Entrance Exams (described in Section 8) task. Although results are still not promising, we already have benchmarks to measure the progress of systems. However, there are still some challenges in QA that still lack proper evaluation. In the following Subsections we gather some observations about how to design evaluations aimed at tackling these main challenges. We include also the problem of answer generation in new architectures, which was not mentioned in Section 8, but introduced in Section 4.

[6] It must be taken into account that a dummy system selecting a candidate answer at random would obtain 0.25.

9.1. Candidate hypotheses generation

The objective of candidate-hypotheses generation is to propose candidate answers given a set of input questions. Then, candidate answers and questions are combined into candidate hypotheses, whose correctness must be proved by the validation component.

Hypotheses generation represents a key task for the whole QA system because if no correct candidate hypothesis is proposed, it is impossible to return a correct answer after the validation process. Therefore, this step must focus on recall rather than precision [4], as in the IR phase of classical QA architectures. That is, this component must generate at least a correct hypothesis. Then, the following steps would add the precision required. However, answer generation has not been widely developed and evaluated in the context of QA systems. Hence, it requires more attention from the QA community.

Although this task is clearly defined, it is not clear how to approach it. Some proposals of the NLP community might be applied to this task as for instance, approaches to the Search Pilot task at the RTE-5, where systems had to return all the sentences in a document entailing a given hypothesis [65]; the vast amount of work on IR; semantic similarity approaches [66]; etc.

We think this step must gather as many potential answers as possible by taking advantage of information from question analysis. Besides, systems should combine several techniques such as text-based search engines applying different approaches, knowledge-bases search, deep-reasoning techniques, etc.

According to the ideas described above, a proper evaluation of this step should encourage the importance of recall over precision, requiring a correct answer among the set of candidates.

9.2. Incorrect answers rejection

We have described in Section 8.3 how the rejection of incorrect answers allows strategies for improving results. For example, systems may find correct answers by discarding the wrong ones in multiple-choice tests.

In order to study the rejection of incorrect answers, we classify incorrect answers into the following three categories:

- *Contradictions with the text*, where an answer contains an incorrect statement according to the meaning of the text.
- *Contradictions with the question*, where the facts stated in an answer are correct according to the text, but incompatible with the expected answer type. For instance, let's take a text containing that a certain event "*X*" took place in a certain place "*Y*" in "*Z*", where "*Z*" refers to a year. If the question asks about the location of the event, the statement "*Event X took place in Z*" is correct according to the text. However, this statement is an incorrect answer to the given question because it does not refer to a location.
- *Unsupported answers*, where the answer might be correct or incorrect, but there is no evidence in the text to decide about its correctness. Here, the validation module should indicate that it is unsure about the answer's correctness.

The ability to classify incorrect answers into these three categories enables QA systems to decide whether: (a) there is a correct answer to a question; (b) there is no answer to a question or, (c) the QA system cannot find a correct answer and leaves a question unanswered. If all the candidate answers are rejected and they belong to any of the first two types, a QA system may decide that there is *no correct answer* to the question. On the other hand, given unsupported answers, the QA system might prefer to leave *unanswered* the question because: (1) it is not able to find a correct answer, or (2) it cannot decide whether there exists a correct answer. Another approach would be to re-configure the generation step and propose new candidates.

We think rejection must receive more attention from QA evaluation campaigns. For instance, QA4MRE 2013 introduced a significant proportion of questions where the *correct answer* was *not provided* among the set of candidates [67]. These questions demanded the ability of detecting that all the candidates were incorrect. However, this was not enough for encouraging the importance of rejection among participant systems.

In order to promote the importance of rejection strategies, we suggest evaluations where the fact of giving incorrect answers would be penalized. This might require the definition of new measures acknowledging the reduction of incorrect answers, with a stronger effect than *c@1*, while maintaining the proportion of correct answers.

9.3. Complex questions

The QA community has already pointed out the difficulty for answering this kind of questions and proposed several approaches aimed at addressing them [68,69]. Nevertheless, the average performance over complex questions remains quite low [70].

From the point of view of evaluation, some campaigns fostered research on complex questions. On the one hand, the Document Understanding Conference (DUC)[7] proposed an evaluation with complex questions, where systems had to extract answers from several documents and combine them into a final summary [71]. However, DUC was mainly focused on automatic summarization instead of QA.

On the other hand, QA4MRE 2013, where test collections contained several complex questions, proposed a pilot study to discover difficulties for answering complex questions [67]. For this purpose, organizers created a simplified version of some questions (a 15% of the whole set), where an inference step was removed. There were three forms of simplification: hypernym replacement, noun phrase synonymy and verbal entailment. For example, given the original question "*What has been offered to the President of the United States if he signs the Kyoto Protocol?*", organizers created the auxiliary question "*What has been offered to Obama if he signs the Kyoto Protocol?*". Results over the simplified questions showed an outstanding increase in performance with respect to the corresponding complex questions (organizers reported more than a 200% of improvement). This observation suggests that systems have problems resolving some inferences required by complex questions. However, the study did not identify what inferences were harder.

Recent advances in textual inference and the use of KBs in NLP might be helpful for answering complex questions, as such KBs have been used for boosting QA performance [72,73]. Moreover, evaluations similar to QA4MRE may encourage better results over complex questions. These evaluations should detect the main inferences contained in complex questions, paying special attention to the most difficult inferences. At the beginning, these evaluations may ask questions requiring few inference steps, where information about the kind of inference is given. Finally, the following stages of these evaluations would propose questions demanding more and harder inference steps.

9.4. Discovering axioms for inference

A proper analysis of documents requires systems to make inferences from the evidences encountered in texts. For instance, let's

[7] http://duc.nist.gov/.

take the text "*It is raining*". It is easy for humans, but not for automatic systems, to infer consequences such as "*wet ground*" or "*people using umbrellas*". However, this is a kind of processing harder than the detection of paraphrases or text reformulations and evaluations have not paid much attention to this challenge, maybe because of its difficulty. We think new evaluations aimed at solving this issue should begin by asking about simple inferences and consequences that cannot be directly extracted from test documents.

This kind of processing might require the joint effort of the KB and QA communities. One promising approach might be the combination of *structured resources* with methods based on the *Open Information Extraction* (OIE) paradigm [74]. While structured resources can give precise information about entities, OIE approaches can complete this knowledge with information from similar entities. These *hybrid approaches* would allow systems to infer the actions usually performed for different professions, roles, classes, etc. For instance, if we see several times in a text collection that different "*goalkeepers*" "*intercepts shots at goal*", we can infer this feature and use it with new goalkeepers.

There are several research questions about these hybrid approaches that might be answered in the context of the suggested evaluations. It is obvious that not all the properties of an entity can be inferred from similar entities, but maybe only some of them. We think future works in QA should be focused on applying these hybrid approaches to make inferences from texts.

10. Related research trends

In previous sections, we have focused on systems developed for evaluation campaigns over free-text. In this section we offer some information about other developments out of the scope of such evaluations

Regarding QA over free-text, some research has focused on improving the performance of QA single modules as a way to improve overall results. Such works proposed advances for question processing [75,76], document retrieval [44], passage selection [77], answer extraction [78] and answer ranking [45,79]. Most of these studies introduced new methods of measuring similarities between questions and answers, relying on techniques aimed to discover deep semantic relationships such as Latent Dirichlet Allocation (LDA) [80], Distributional Semantic Models (DSMs) [45] and Latent Semantic Analysis [81], etc.

There have been also some works focused on developing QA technologies for restricted domains such as legal [82] and biomedicine [83] documents. In the case of the biomedicine domain, it attracted such interest that there was an evaluation campaign focus on it [24].

Other researchers have also worked on scaling QA systems over free-text to the Web [84,85]. The objective of these systems is to make the web more accessible to common users taking advantage of QA technologies. However, these systems find difficulties such as different versions of the same fact, a not so controlled language, typos, etc. This is why some systems rely on knowledge bases to overcome such difficulties [72].

Another area of interest is to exploit the data contained in community QA sites such as Yahoo! Answers[8], where users ask and answer questions about different topics [86]. These sites become a huge repository of human answers to questions formulated by users. While some researchers focused on detecting the topics for each question [87], others worked about evaluating the effectiveness of such sites by assessing user satisfaction [88]. Furthermore, since 2015 there has been an evaluation task at SemEval for community Question Answering given the growing interest in this area [89,90].

On the other hand, we have observed advances in semantic search and QA over RDF data in recent years. These approaches focused on the use of semantics to improve search results especially over the Web [91] and open-domain knowledge sources [92,93], with some emphasis on structured data [94,95] and linked data [96,97]. These systems must deal with the management of structured and semantic data, as well as the problem of the lexical gap between the vocabulary of users and that of semantic resources. This interest on QA over RDF motivated the proposal of evaluation campaigns focused on offering a common framework for testing systems. Examples of such campaigns are SEALS [98] for semantic search and QALD for Natural Language interfaces to structure data, with special emphasis on Linked Data [99]. Although these evaluations worked apart from free-text evaluations, we think they might converge in the future.

11. Conclusions

In this paper, we have analyzed how Question Answering (QA) benchmarking affected the development of this technology. After the inclusion of Information Retrieval (IR) modules, QA systems achieved big advances in results. However, QA systems rapidly reached an upper-bound performance close to 60% of questions correctly answered. The QA community suggested the dependence on the IR component and the use of pipeline architectures, highly affected by error propagation, as the main reasons for such upper bound. These observations encouraged some proposals towards the development of new architectures based on the generation of answers as hypotheses and their validation.

Some of these proposals relied on counting redundancies among answers, while other approaches applied Machine Learning techniques. Nevertheless, systems did not achieve the expected results. This is why the community proposed the evaluation of Reading Comprehension (RC) tests as a strategy to push forward the development of validation technologies, with the objective of improving QA performance. But, again, the results were quite poor.

With the purpose of detecting the main reasons for such low performance, in this paper we have analyzed some RC collections in detail. In this study, we have paid special attention to those questions where systems obtained the worst results. The main conclusion of this study is that QA systems fail on questions requiring a complex analysis of texts. This is because, in RC tests, systems usually find challenges such as different rewordings in questions and source documents, information not explicitly mentioned in documents, etc. However, current systems have adopted naive strategies based on measuring similarities that are insufficient for solving RC tests.

After this study, we have realized that benchmarks for evaluating the final performance of systems using RC tests are a good starting point. However, the challenge now is to balance this kind of extrinsic evaluation with tasks able to provide more insights for the development of QA technologies. This is why we finally offer some directions of evaluations aimed at overcoming the main challenges for current QA technologies and obtaining promising results in a short period of time. Future work is oriented at proposing and developing such evaluations inside the QA community.

Acknowledgment

This work has been partially supported by the Spanish Government (MINECO) in the framework TUNER TIN 2015-65308-C5-1-R (MINECO/FEDER, UE) project.

[8] answers.yahoo.com.

References

[1] B. Magnini, D. Giampiccolo, P. Forner, C. Ayache, V. Jijkoun, P. Osenova, A. Peñas, P. Rocha, B. Sacaleanu, R.F.E. Sutcliffe, Overview of the CLEF 2006 multilingual question answering track, in: 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. LNCS 4730, 2007, pp. 223–256.

[2] A. Peñas, P. Forner, R. Sutcliffe, A. Rodrigo, C. Forăscu, I.n. Alegria, D. Giampiccolo, N. Moreau, P. Osenova, Overview of ResPubliQA 2009: question answering evaluation over European Legislation, in: Proceedings of the 10th cross-language evaluation forum conference on Multilingual information access evaluation: text retrieval experiments, 2009, pp. 174–196.

[3] J. Weston, A. Bordes, S. Chopra, A.M. Rush, B. van Merrienboer, A. Joulin, T. Mikolov, Towards ai-complete question answering: A set of prerequisite toy tasks, 2015 arXiv preprint arXiv:1502.05698.

[4] D.A. Ferrucci, E.W. Brown, J. Chu-Carroll, J. Fan, D. Gondek, A. Kalyanpur, A. Lally, J.W. Murdock, E. Nyberg, J.M. Prager, N. Schlaefer, C.A. Welty, Building watson: an overview of the DeepQA project, AI Mag. 31 (3) (2010) 59–79.

[5] C. Cleverdon, The Cranfield tests on index language devices, in: Aslib Proceedings, vol. 19, 1967, pp. 173–192.

[6] E.M. Voorhees, Overview of the TREC 2003 question answering track, in: Proceedings of the Twelfth Text REtrieval Conference (TREC 2003), 2003, pp. 54–68.

[7] E.M. Voorhees, D.M. Tice, The TREC-8 question answering track evaluation, in: Text Retrieval Conference TREC-8, 1999, pp. 83–105.

[8] E.M. Voorhees, Overview of TREC 2002 question answering track, in: E.M. Voorhees, L.P. Buckland (Eds.), Proceedings of the Eleventh Text REtrieval Conference (TREC 2002). NIST Publication 500-251, 2002, pp. 57–68.

[9] J. Herrera, A. Peñas, F. Verdejo, Question answering pilot task at CLEF 2004, in: C. Peters, J. Gonzalo, M. Kluck, P. Clough, G. Jones, B. Magnini (Eds.), Multilingual Information Access for Text, Speech and Images. CLEF 2004, Lecture Notes in Computer Science, vol. 3491, 2005, pp. 581–590.

[10] A. Peñas, A. Rodrigo, A simple measure to assess non-response, in: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1, in: HLT '11, Association for Computational Linguistics, 2011, pp. 1415–1424.

[11] E. Brown, E. Epstein, J.W. Murdock, T.-H. Fin, Tools and methods for building watson, IBM Res. 14 (2013) 2013.

[12] M.-D. Olvera-Lobo, J. Gutiérrez-Artacho, Question answering track evaluation in TREC, CLEF and NTCIR, in: New Contributions in Information Systems and Technologies: Volume 1, 2015, pp. 13–22.

[13] M. Potthast, T. Gollub, F.M.R. Pardo, P. Rosso, E. Stamatatos, B. Stein, Improving the reproducibility of PAN's shared tasks: plagiarism detection, author identification, and author profiling, in: Information Access Evaluation. Multilinguality, Multimodality, and Interaction - 5th International Conference of the CLEF Initiative, CLEF 2014, Sheffield, UK, September 15–18, 2014. Proceedings, 2014, pp. 268–299.

[14] B.F. Green Jr., A.K. Wolf, C. Chomsky, K. Laughery, Baseball: an automatic question-answerer, in: Papers Presented at the May 9–11, 1961, Western Joint IRE-AIEE-ACM Computer Conference, in: IRE-AIEE-ACM '61 (Western), ACM, New York, NY, USA, 1961, pp. 219–224.

[15] W.A. Woods, Progress in natural language understanding: an application to lunar geology, in: Proceedings of the June 4–8, 1973, National Computer Conference and Exposition, in: AFIPS '73, ACM, New York, NY, USA, 1973, pp. 441–450.

[16] R. Wilensky, D.N. Chin, M. Luria, J. Martin, J. Mayfield, D. Wu, The Berkeley unix consultant project, Comput. Linguist. 14 (4) (1988) 35–84.

[17] O. Herzog, C.-R. Rollinger (Eds.), Text Understanding in LILOG, Integrating Computational Linguistics and Artificial Intelligence, Final Report on the IBM Germany LILOG-Project, vol. vol. 546Lecture Notes in Computer Science, Springer, 1991.

[18] C.D. Manning, P. Raghavan, H. Schütze, Introduction to Information Retrieval, Cambridge University Press, 2008.

[19] E. Voorhees, D. Harman, N.I. of Standards, T. (US), TREC: Experiment and Evaluation in Information Retrieval, vol. 63, MIT press, Cambridge MA, 2005.

[20] D. Moldovan, S. Harabagiu, M. Pasca, R. Mihalcea, R. Girju, R. Goodrum, V. 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, 2000, pp. 563–570.

[21] E. Hovy, L. Gerber, U. Hermjakob, M. Junk, C.-Y. Lin, Question answering in webclopedia, in: Proceedings of the Ninth Text REtrieval Conference, 2001, pp. 655–664.

[22] J. Prager, E. Brown, A. Coden, D.R. Radev, Question-answering by predictive annotation, in: Proceedings of the 23rd SIGIR Conference, 2000, pp. 184–191.

[23] E. Agichtein, D. Carmel, D. Pelleg, Y. Pinter, D. Harman, Overview of the TREC 2015 LiveQA track, in: Proceedings of The Twenty-Fourth Text REtrieval Conference, TREC 2015, Gaithersburg, Maryland, USA, November 17–20, 2015, 2015.

[24] G. Tsatsaronis, G. Balikas, P. Malakasiotis, I. Partalas, M. Zschunke, M.R. Alvers, D. Weissenborn, A. Krithara, S. Petridis, D. Polychronopoulos, Y. Almirantis, J. Pavlopoulos, N. Baskiotis, P. Gallinari, T. Artières, A. Ngonga, N. Heino, É. Gaussier, L. Barrio-Alvers, M. Schroeder, I. Androutsopoulos, G. Paliouras, An overview of the BIOASQ large-scale biomedical semantic indexing and question answering competition, BMC Bioinform. 16 (2015) 138:1–138:28.

[25] D. Sonntag, Distributed NLP and machine learning for question answering grid, in: Proceedings of the Workshop on Semantic Intelligent Middleware for the Web and the Grid at the 16th European Conference on Artificial Intelligence (ECAI), 2004.

[26] I. Dagan, O. Glickman, B. Magnini, The PASCAL recognising textual entailment challenge, in: J.Q. Candela, I. Dagan, B. Magnini, F. d'Alché Buc (Eds.), MLCW 2005, LNAI, Lecture Notes in Computer Science, vol. 3944, Springer, 2005, pp. 177–190.

[27] B. Magnini, M. Negri, R. Prevete, H. Tanev, Is it the right answer? Exploiting web redundancy for answer validation, in: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, July, 2002, pp. 425–432.

[28] S. Harabagiu, A. Hickl., Methods for using textual entailment in open-domain question answering, in: Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL, Sydney, 2006, pp. 905–912.

[29] E. Brill, J.J. Lin, M. Banko, S.T. Dumais, A.Y. Ng, Data-intensive question answering, in: Proceedings of the Tenth Text Retrieval Conference (TREC), 2001, pp. 393–400.

[30] M. Tonoike, T. Utsuro, S. Sato, Answer validation by keyword association, in: Proc. of the 3rd Workshop on Robust Methods in Analysis of Natural Language Data, 2004.

[31] W. Bosma, C. Callison-Burch, Paraphrase substitution for recognizing textual entailment, in: 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: LNCS 4730, 2007, pp. 502–509.

[32] A. Iftene, Building a textual entailment system for the RTE3 competition. Application to a QA system, in: Proceedings of the 2008 10th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, in: SYNASC '08, IEEE Computer Society, Washington, DC, USA, 2008, pp. 116–122.

[33] S. Harabagiu, S. Maiorano, Finding answers in large collections of texts: paragraph indexing+ abductive inference, in: Proceedings of the AAAI Fall Symposium on Question Answering Systems, 1999, pp. 63–71.

[34] D.I. Moldovan, S.M. Harabagiu, R. Girju, P. Morarescu, V.F. Lacatusu, A. Novischi, A. Badulescu, O. Bolohan, LCC tools for question answering, in: Eleventh Text REtrieval Conference (TREC-2002), 2002, pp. 144–154.

[35] M. Tatu, B. Iles, D. Moldovan, Automatic answer validation using COGEX, in: C. Peters, P. Clough, F. Gey, J. Karlgren, B. Magnini, D. Oard, M. Rijke, M. Stempfhuber (Eds.), Evaluation of Multilingual and Multi-modal Information Retrieval, Lecture Notes in Computer Science, vol. 4730, Springer Berlin Heidelberg, 2007, pp. 494–501.

[36] I. Glöckner, RAVE: a fast logic-based answer validator, in: Evaluating Systems for Multilingual and Multimodal Information Access: 9th Workshop of the Cross-Language Evaluation Forum, CLEF 2008, Revised Selected Papers, in: LNCS 5706, 2009, pp. 468–471.

[37] A. Peñas, Á. Rodrigo, V. Sama, F. Verdejo, Testing the reasoning for question answering validation, J. Logic Comput. 18 (3) (2008) 459–474.

[38] E.J. Breck, J.D. Burger, L. Ferro, L. Hirschman, D. House, M. Light, I. Mani, How to evaluate your question answering system every day and still get real work done, in: Proceedings of the Second Conference on Language Resources and Evaluation (LREC-2000), 2000, pp. 1495–1500.

[39] Á. Rodrigo, A. Peñas, F. Verdejo, Overview of the answer validation exercise 2008, in: Evaluating Systems for Multilingual and Multimodal Information Access, 9th Workshop of the Cross-Language Evaluation Forum, CLEF 2008, Revised Selected Papers CLEF, Springer, 5706, 2009, pp. 296–313.

[40] M. Day, C. Tsai, A study on machine learning for imbalanced datasets with answer validation of question answering, in: 17th IEEE International Conference on Information Reuse and Integration, IRI 2016, Pittsburgh, PA, USA, July 28–30, 2016, 2016, pp. 513–519.

[41] I. Glöckner, Towards logic-based question answering under time constraints, in: Proceedings of the International Multi-Conference of Engineers and Computer Scientists, 2008, pp. 13–18.

[42] A. Téllez-Valero, M. Montes-Y-Gómez, L. Villaseñor-Pineda, A. Peñas, Improving question answering by combining multiple systems via answer validation, in: A.F. Gelbukh (Ed.), CICLing, Lecture Notes in Computer Science, vol. 4919, Springer, 2008, pp. 544–554.

[43] Á. Rodrigo, A. Peñas, On evaluating the contribution of validation for question answering, IEEE Trans. Knowl. Data Eng. 27 (4) (2015) 1157–1161.

[44] J. Pérez-Iglesias, G. Garrido, Á. Rodrigo, L. Araujo, A. Peñas, Information retrieval baselines for the ResPubliQA task, in: Multilingual Information Access Evaluation I. Text Retrieval Experiments, 10th Workshop of the Cross-Language Evaluation Forum, CLEF 2009, Corfu, Greece, September 30, - October 2, 2009, Revised Selected Papers, 2009, pp. 253–256.

[45] P. Molino, Semantic models for answer re-ranking in question answering, in: Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, in: SIGIR '13, 2013. 1146–1146

[46] N. Othman, R. Faiz, A multi-lingual approach to improve passage retrieval for automatic question answering, in: Natural Language Processing and Information Systems - 21st International Conference on Applications of Natural Language to Information Systems, NLDB 2016, Salford, UK, June 22–24, 2016, Proceedings, 2016, pp. 127–139.

[47] P. Rajpurkar, J. Zhang, K. Lopyrev, P. Liang, SQuAD: 100, 000+ questions for machine comprehension of text, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1–4, 2016, 2016, pp. 2383–2392.

[48] M. Richardson, C.J.C. Burges, E. Renshaw, MCTest: a challenge dataset for the open-domain machine comprehension of text, in: Proceedings of the 2013

Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18–21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL, 2013, pp. 193–203.
[49] A. Peñas, E. Hovy, P. Forner, Á. Rodrigo, R. Sutcliffe, C. Forascu, C. Sporleder, Evaluating machine reading systems through comprehension tests, in: Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12), Istanbul, Turkey, 2012, pp. 1143–1147.
[50] P. Clark, O. Etzioni, My computer is an honor student - but how intelligent is it? Standardized tests as a measure of AI, AI Mag. 37 (1) (2016) 5–12.
[51] J. Pustejovsky, The syntax of event structure, Cognition 41 (1) (1991) 47–81.
[52] R. Datta, D. Joshi, J. Li, J.Z. Wang, Image retrieval: ideas, influences, and trends of the new age, ACM Comput. Surv. (CSUR) 40 (2) (2008) 5.
[53] P. Shvaiko, J. Euzenat, A survey of schema-based matching approaches, J. Data Semant. IV (2005) 146–171. Springer
[54] D. Gildea, D. Jurafsky, Automatic labeling of semantic roles, Comput. Linguist. 28 (3) (2002) 245–288.
[55] R. Bar-Haim, I. Dagan, B. Dolan, L. Ferro, D. Giampiccolo, B. Magnini, I. Szpektor, The second PASCAL recognising textual entailment challenge, in: Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment, Venezia, Italy, 2006, pp. 1–9.
[56] D. Lin, P. Pantel, Discovery of inference rules for question-answering, Nat. Lang. Eng. 7 (4) (2001) 343–360.
[57] N. Guarino, C. Masolo, G. Vetere, OntoSeek: content-Based access to the web, IEEE Intell. Syst. 14 (3) (1999) 70–80.
[58] A. Peñas, Y. Miyao, E. Hovy, P. Forner, N. Kando, Overview of QA4MRE 2013 Entrance Exams Task, CLEF, 2013. Working Notes
[59] A.M. Turing, Computing Machinery and Intelligence, 1950.
[60] S. Harabagiu, F. Lacatusu, A. Hickl, Answering complex questions with random walk models, in: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, in: SIGIR '06, 2006, pp. 220–227.
[61] E. Saquete, P. Martínez-Barco, R. Muñoz, J.L. Vicedo, Splitting complex temporal questions for question answering systems, in: Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics, in: ACL '04, 2004.
[62] D. Moldovan, M. Paşca, S. Harabagiu, M. Surdeanu, Performance issues and error analysis in an open-domain question answering system, ACM Trans. Inf. Syst. 21 (2) (2003) 133–154.
[63] P. Clark, O. Etzioni, T. Khot, A. Sabharwal, O. Tafjord, P.D. Turney, D. Khashabi, Combining retrieval, statistics, and inference to answer elementary science questions, in: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12–17, 2016, Phoenix, Arizona, USA., 2016, pp. 2580–2586.
[64] D. Ferrucci, E. Nyberg, J. Allen, K. Barker, E.W. Brown, J. Chu-Carroll, A. Ciccolo, P.A. Duboue, J. Fan, D. Gondek, E. Hovy, B. Katz, A. Lally, M. McCord, P. Morarescu, J.W. Murdock, B. Porter, J.M. Prager, T. Strzalkowski, C. Welty, W. Zadrozny, Towards the Open Advancement of Question Answering Systems, Technical Report, IBM Research, Hawhtorne, NY, 2008.
[65] L. Bentivogli, I. Dagan, H.T. Dang, D. Giampiccolo, B. Magnini, The fifth PASCAL recognizing textual entailment challenge, in: Proceedings of the Text Analysis Conference (TAC09), 2009.
[66] E. Agirre, D. Cer, M. Diab, A. Gonzalez-Agirre, W. Guo, SEM 2013 shared task: semantic textual similarity, in: Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity, Association for Computational Linguistics, 2013, pp. 32–43.
[67] R. Sutcliffe, A. Peñas, E. Hovy, P. Forner, Á. Rodrigo, C. Forascu, Y. Benajiba, P. Osenova, Overview of QA4MRE Main Task at CLEF 2013, CLEF, 2013. Working Notes
[68] S. Narayanan, S. Harabagiu, Question answering based on semantic structures, in: Proceedings of the 20th International Conference on Computational Linguistics, in: COLING '04, 2004.
[69] R. Soricut, E. Brill, Automatic question answering using the web: beyond the factoid, Inf. Retr. 9 (2) (2006) 191–206.
[70] J. Weston, S. Chopra, A. Bordes, Memory networks, Int. Conf. Learn. Represent. (ICLR) (2015).
[71] H.T. Dang, Overview of DUC 2006, in: Proceedings of HLT-NAACL 2006, 2006.
[72] H. Sun, H. Ma, W.-t. Yih, C.-T. Tsai, J. Liu, M.-W. Chang, Open domain question answering via semantic enrichment, in: Proceedings of the 24th International Conference on World Wide Web, in: WWW '15, 2015, pp. 1045–1055.
[73] W. Yih, M. Chang, C. Meek, A. Pastusiak, Question answering using enhanced lexical semantic models, in: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4–9 August 2013, Sofia, Bulgaria, Volume 1: Long Papers, 2013, pp. 1744–1753.
[74] O. Etzioni, M. Banko, S. Soderland, D.S. Weld, Open information extraction from the web, Commun. ACM 51 (12) (2008) 68–74.
[75] Z. Hui, J. Liu, L. Ouyang, Question classification based on an extended class sequential rule model, in: Proceedings of 5th International Joint Conference on Natural Language Processing, Asian Federation of Natural Language Processing, Chiang Mai, Thailand, 2011, pp. 938–946.
[76] S. Xu, G. Cheng, F. Kong, Research on question classification for automatic question answering, in: 2016 International Conference on Asian Language Processing, IALP 2016, Tainan, Taiwan, November 21–23, 2016, 2016, pp. 218–221.
[77] M. Tan, C. dos Santos, B. Xiang, B. Zhou, Improved representation learning for question answer matching, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Berlin, Germany, 2016, pp. 464–473.
[78] H. Sun, N. Duan, Y. Duan, M. Zhou, Answer extraction from passage graph for question answering, in: Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, in: IJCAI '13, 2013, pp. 2169–2175.
[79] S.-J. Yen, Y.-C. Wu, J.-C. Yang, Y.-S. Lee, C.-J. Lee, J.-J. Liu, A support vector machine-based context-ranking model for question answering, Inf. Sci. 224 (0) (2013) 77–87.
[80] A. Celikyilmaz, D. Hakkani-Tur, G. Tur, LDA based similarity modeling for question answering, in: Proceedings of the NAACL HLT 2010 Workshop on Semantic Search, in: SS '10, 2010, pp. 1–9.
[81] A. Gliozzo, Latent Semantic Analysis for Application in a Question Answer System, 2014, US Patent App. 13/765,323.
[82] M.-Y. Kim, Y. Xu, R. Goebel, Applying a convolutional neural network to legal question answering, in: M. Otake, S. Kurahashi, Y. Ota, K. Satoh, D. Bekki (Eds.), New Frontiers in Artificial Intelligence: JSAI-isAI 2015 Workshops, LENLS, JURISIN, AAA, HAT-MASH, TSDAA, ASD-HR, and SKL, Kanagawa, Japan, November 16–18, 2015, Revised Selected Papers, 2017, pp. 282–294.
[83] M. Neves, U. Leser, Question answering for biology, Methods 74 (2015) 36–46.
[84] J. Lehmann, T. Furche, G. Grasso, A.N. Ngomo, C. Schallhart, A.J. Sellers, C. Unger, L. Bühmann, D. Gerber, K. Höffner, D. Liu, S. Auer, Deqa: deep web extraction for question answering, in: The Semantic Web - ISWC 2012 - 11th International Semantic Web Conference, Boston, MA, USA, November 11–15, 2012, Proceedings, Part II, 2012, pp. 131–147.
[85] S. Verberne, L. Boves, W. Kraaij, Bringing why-QA to web search, in: Proceedings of the 33rd European Conference on Advances in Information Retrieval, in: ECIR'11, 2011, pp. 491–496.
[86] A. Shtok, G. Dror, Y. Maarek, I. Szpektor, Learning from the past: answering new questions with past answers, in: Proceedings of the 21st International Conference on World Wide Web, in: WWW '12, 2012, pp. 759–768.
[87] L. Cai, G. Zhou, K. Liu, J. Zhao, Learning the latent topics for question retrieval in community QA, in: Proceedings of 5th International Joint Conference on Natural Language Processing, 2011.
[88] Y. Liu, J. Bian, E. Agichtein, Predicting information seeker satisfaction in community question answering, in: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, in: SIGIR '08, 2008, pp. 483–490.
[89] P. Nakov, L. Màrquez, W. Magdy, A. Moschitti, J. Glass, B. Randeree, Semeval-2015 task 3: answer selection in community question answering, in: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), 2015, pp. 269–281.
[90] P. Nakov, L. Màrquez, A. Moschitti, W. Magdy, H. Mubarak, A.A. Freihat, J. Glass, B. Randeree, SemEval-2016 task 3: community question answering, in: Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16–17, 2016, 2016, pp. 525–545.
[91] R. Guha, R. McCool, E. Miller, Semantic search, in: Proceedings of the 12th International Conference on World Wide Web, in: WWW '03, ACM, New York, NY, USA, 2003, pp. 700–709.
[92] A. Fader, L. Zettlemoyer, O. Etzioni, Paraphrase-driven learning for open question answering, in: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2013, pp. 1608–1618.
[93] A. Fader, L. Zettlemoyer, O. Etzioni, Open question answering over curated and extracted knowledge bases, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, in: KDD '14, 2014, pp. 1156–1165.
[94] L. Zou, R. Huang, H. Wang, J.X. Yu, W. He, D. Zhao, Natural language question answering over RDF: a graph data driven approach, in: Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data, in: SIGMOD '14, 2014, pp. 313–324.
[95] H. Li, Y. Wang, G. de Melo, C. Tu, B. Chen, Multimodal question answering over structured data with ambiguous entities, in: Proceedings of the 26th International Conference on World Wide Web Companion, in: WWW '17 Companion, 2017, pp. 79–88.
[96] M. Yahya, K. Berberich, S. Elbassuoni, G. Weikum, Robust question answering over the web of linked data, in: Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management, in: CIKM '13, 2013, pp. 1107–1116.
[97] C. Unger, A. Freitas, P. Cimiano, An introduction to question answering over linked data, in: Reasoning Web. Reasoning on the Web in the Big Data Era - 10th International Summer School 2014, Athens, Greece, September 8–13, 2014. Proceedings, 2014, pp. 100–140.
[98] S.N. Wrigley, K. Elbedweihy, D. Reinhardt, A. Bernstein, F. Ciravegna, Evaluating semantic search tools using the SEALS platform, International Workshop on Evaluation of Semantic Technologies (IWEST 2010) Workshop, 2010.
[99] V. Lopez, C. Unger, P. Cimiano, E. Motta, Evaluating question answering over linked data, Web Semant. 21 (0) (2013).