A Simple Measure to Assess Non-response

Anselmo Peñas and Alvaro Rodrigo
UNED NLP & IR Group
Juan del Rosal, 16
28040 Madrid, Spain
{anselmo,alvarory@lsi.uned.es}

Abstract

There are several tasks where is preferable not responding than responding incorrectly. This idea is not new, but despite several previous attempts there isn't a commonly accepted measure to assess non-response. We study here an extension of *accuracy* measure with this feature and a very easy to understand interpretation. The measure proposed (*c@1*) has a good balance of discrimination power, stability and sensitivity properties. We show also how this measure is able to reward systems that maintain the same number of correct answers and at the same time decrease the number of incorrect ones, by leaving some questions unanswered. This measure is well suited for tasks such as Reading Comprehension tests, where multiple choices per question are given, but only one is correct.

1 Introduction

There is some tendency to consider that an incorrect result is simply the absence of a correct one. This is particularly true in the evaluation of Information Retrieval systems where, in fact, the absence of results sometimes is the worse output.

However, there are scenarios where we should consider the possibility of not responding, because this behavior has more value than responding incorrectly. For example, during the process of introducing new features in a search engine it is important to preserve users' confidence in the system. Thus, a system must decide whether it should give or not a result in the new fashion or keep on with the old kind of output. A similar example is the decision about showing or not ads related to the query. Showing wrong ads harms the business model more than showing nothing. A third example more related to Natural Language Processing is the Machine Reading evaluation through reading comprehension tests. In this case, where multiple choices for a question are offered, choosing a wrong option should be punished against leaving the question unanswered.

In the latter case, the use of utility functions is a very common option. However, utility functions give arbitrary value to not responding and ignore the system's behavior showed when it responds (see Section 2). To avoid this, we present c@1 measure (Section 2.2), as an extension of accuracy (the proportion of correctly answered questions). In Section 3 we show that no other extension produces a sensible measure. In Section 4 we evaluate c@1 in terms of stability, discrimination power and sensibility, and some real examples of its behavior are given in the context of Question Answering. Related work is discussed in Section 5.

2 Looking for the Value of Not Responding

Lets take the scenario of Reading Comprehension tests to argue about the development of the measure. Our scenario assumes the following:

- There are several questions.
- Each question has several options.
- One option is correct (and only one).

The first step is to consider the possibility of not responding. If the system responds, then the assessment will be one of two: correct or wrong. But if

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, pages 1415–1424,
Portland, Oregon, June 19-24, 2011.

the system doesn't respond there is no assessment. Since every question has a correct answer, non response is not correct but it is not incorrect either. This is represented in contingency Table 1, where:

- n_{ac}: number of questions for which the answer is correct
- n_{aw}: number of questions for which the answer is incorrect
- n_u: number of questions not answered
- n: number of questions ($n = n_{ac} + n_{aw} + n_u$)

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

Table 1: Contingency table for our scenario

Let's start studying a simple utility function able to establish the preference order we want:

- -1 if question receives an incorrect response
- 0 if question is left unanswered
- 1 if question receives a correct response

Let *U(i)* be the utility function that returns one of the above values for a given question *i*. Thus, if we want to consider *n* questions in the evaluation, the measure would be:

$$UF = \frac{1}{n}\sum_{i=1}^{n} U(i) = \frac{n_{ac} - n_{aw}}{n} \qquad (1)$$

The rationale of this utility function is intuitive: not answering adds no value and wrong answers add negative values. Positive values of *UF* indicate more correct answers than incorrect ones, while negative values indicate the opposite. However, the utility function is giving an arbitrary value to the preferences (-1, 0, 1).

Now we want to interpret in some way the value that Formula (1) assigns to unanswered questions. For this purpose, we need to transform Formula (1) into a more meaningful measure with a parameter for the number of unanswered questions (n_u). A monotonic transformation of (1) permit us to preserve the ranking produced by the measure. Let f(x)=0.5x+0.5 be the monotonic function to be used for the transformation. Applying this function to Formula (1) results in Formula (2):

$$\begin{aligned} 0.5\frac{n_{ac} - n_{aw}}{n} + 0.5 &= \frac{0.5}{n}\left[n_{ac} - n_{aw} + n\right] = \\ &= \frac{0.5}{n}\left[n_{ac} - n_{aw} + n_{ac} + n_{aw} + n_u\right] \\ &= \frac{0.5}{n}\left[2n_{ac} + n_u\right] = \frac{n_{ac}}{n} + 0.5\frac{n_u}{n} \end{aligned} \qquad (2)$$

Measure (2) provides the same ranking of systems than measure (1). The first summand of Formula (2) corresponds to *accuracy*, while the second is adding an arbitrary constant weight of 0.5 to the proportion of unanswered questions. In other words, *unanswered questions are receiving the same value as if half of them had been answered correctly*.

This does not seem correct given that not answering is being rewarded in the same proportion to all the systems, without taking into account the performance they have shown with the answered questions. We need to propose a more sensible estimation for the weight of unanswered questions.

2.1 A rationale for the Value of Unanswered Questions

According to the utility function suggested, unanswered questions would have value as if half of them had been answered correctly. Why half and not other value? Even more, Why a constant value? Let's generalize this idea and estate more clearly our hypothesis:

Unanswered questions have the same value as if a proportion of them would have been answered correctly.

We can express this idea according to contingency Table 1 in the following way:

$$\begin{aligned} P(C) &= P(C \cap A) + P(C \cap \neg A) = \\ &= P(C \cap A) + P(C/\neg A) * P(\neg A) \end{aligned} \qquad (3)$$

$P(C \cap A)$ can be estimated by n_{ac}/n, $P(\neg A)$ can be estimated by n_u/n, and we have to estimate $P(C/\neg A)$. Our hypothesis is saying that $P(C/\neg A)$

is different from 0. The utility measure (2) corresponds to P(C) in Formula (3) where $P(C/\neg A)$ receives a constant value of 0.5. It is assuming arbitrarily that $P(C/\neg A) = P(C/A)$.

Following this, our measure must consist of two parts: The overall *accuracy* and a better estimation of correctness over the unanswered questions.

2.2 The Measure Proposed: c@1

From the answered questions we have already observed the proportion of questions that received a correct answer ($P(C \cap A)$ = n_{ac}/n). We can use this observation as our estimation for $P(C/\neg A)$ instead of the arbitrary value of 0.5.

Thus, the measure we propose is c@1 (correctness at one) and is formally represented as follows:

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

The most important features of c@1 are:

1. A system that answers all the questions will receive a score equal to the traditional *accuracy* measure: n_u=0 and therefore *c@1*=n_{ac}/n.

2. Unanswered questions will add value to *c@1* as if they were answered with the *accuracy* already shown.

3. A system that does not return any answer would receive a score equal to 0 due to n_{ac}=0 in both summands.

According to the reasoning above, we can interpret *c@1* in terms of probability as $P(C)$ where $P(C/\neg A)$ has been estimated with $P(C \cap A)$. In the following section we will show that there is no other estimation for $P(C/\neg A)$ able to provide a reasonable evaluation measure.

3 Other Estimations for $P(C/\neg A)$

In this section we study whether other estimations of $P(C/\neg A)$ can provide a sensible measure for QA when unanswered questions are taken into account. They are:

1. $P(C/\neg A) \equiv 0$
2. $P(C/\neg A) \equiv 1$
3. $P(C/\neg A) \equiv P(\neg C/\neg A) \equiv 0.5$
4. $P(C/\neg A) \equiv P(C/A)$
5. $P(C/\neg A) \equiv P(\neg C/A)$

3.1 $P(C/\neg A) \equiv 0$

This estimation considers the absence of response as incorrect response and we have the traditional *accuracy* (n_{ac}/n).

Obviously, this is against our purposes.

3.2 $P(C/\neg A) \equiv 1$

This estimation considers all unanswered questions as correctly answered. This option is not reasonable and is given for completeness: systems giving no answer would get maximum score.

3.3 $P(C/\neg A) \equiv P(\neg C/\neg A) \equiv 0.5$

It could be argued that since we cannot have observations of correctness for unanswered questions, we should assume equiprobability between $P(C/\neg A)$ and $P(\neg C/\neg A)$. In this case, P(C) corresponds to the expression (2) already discussed. As previously explained, in this case we are giving an arbitrary constant value to unanswered questions independently of the system's performance shown with answered ones. This seems unfair. We should be aiming at rewarding those systems not responding instead of giving wrong answers, not reward the sole fact that the system is not responding.

3.4 $P(C/\neg A) \equiv P(C/A)$

An alternative is to estimate the probability of correctness for the unanswered questions as the precision observed over the answered ones: P(C/A)= n_{ac}/(n_{ac}+ n_{aw}). In this case, our measure would be like the one shown in Formula (5):

$$\begin{aligned} P(C) &= P(C \cap A) + P(C/\neg A) * P(\neg A) = \\ &= P(C/A) * P(A) + P(C/A) * P(\neg A) = \\ &= P(C/A) = \frac{n_{ac}}{n_{ac} + n_{aw}} \end{aligned} \quad (5)$$

The resulting measure is again the observed precision over the answered ones. This is not a sensible measure, as it would reward a cheating system that decides to leave all questions unanswered except one for which it is sure to have a correct answer.

Furthermore, from the idea that $P(C/\neg A)$ is equal to $P(C/A)$ the underlying assumption is that systems choose to answer or not to answer randomly, whereas we want to reward the systems that choose not responding because they are able to decide that their candidate options are wrong or because they are unable to decide which candidate is correct.

3.5 $P(C/\neg A) \equiv P(\neg C/A)$

The last option to be considered explores the idea that systems fail not responding in the same proportion that they fail when they give an answer (i.e. proportion of incorrect answers).

Estimating $P(C/\neg A)$ as n_{aw} / (n_{ac}+ n_{aw}), the measure would be:

$$\begin{aligned} P(C) &= P(C \cap A) + P(C/\neg A) * P(\neg A) = \\ &= P(C \cap A) * P(\neg C/A) * P(\neg A) = \\ &= \frac{n_{ac}}{n} + \frac{n_{aw}}{n_{ac} + n_{aw}} * \frac{n_u}{n} \end{aligned} \quad (6)$$

This measure is very easy to cheat. It is possible to obtain almost a perfect score just by answering incorrectly only one question and leaving unanswered the rest of the questions.

4 Evaluation of c@1

When a new measure is proposed, it is important to study the reliability of the results obtained using that measure. For this purpose, we have chosen the method described by Buckley and Voorhees (2000) for assessing the stability and discrimination power, as well as the method described by Voorhees and Buckley (2002) for examining the sensitivity of our measure. These methods have been used for studying IR metrics (showing similar results with the methods based on statistics (Sakai, 2006)), as well as for evaluating the reliability of other QA measures different to the ones studied here (Sakai, 2007a; Voorhees, 2002; Voorhees, 2003).

We have compared the results over c@1 with the ones obtained using both *accuracy* and the *utility function* (UF) defined in Formula (1). This comparison is useful to show how confident can a researcher be with the results obtained using each evaluation measure.

In the following subsections we will first show the data used for our study. Then, the experiments about stability and sensitivity will be described.

4.1 Data sets

We used the test collections and runs from the Question Answering track at the Cross Language Evaluation Forum 2009 (CLEF) (Peñas et al., 2010). The collection has a set of 500 questions with their answers. The 44 runs in different languages contain the human assessments for the answers given by actual participants. Systems could chose not to answer a question. In this case, they had the chance to submit their best candidate in order to assess the performance of their validation module (the one that decides whether to give or not the answer).

This data collection allows us to compare *c@1* and *accuracy* over the same runs.

4.2 Stability vs. Discrimination Power

The more stable a measure is, the lower the probability of errors associated with the conclusion *"system A is better than system B"* is. Measures with a high error must be used more carefully performing more experiments than in the case of using a measure with lower error.

In order to study the stability of c@1 and to compare it with *accuracy* we used the method described by Buckley and Voorhees (2000). This method allows also to study the number of times systems are deemed to be equivalent with respect to a certain measure, which reflects the *discrimination power* of that measure. The less discriminative the measure is, the more ties between systems there will be. This means that longer difference in scores will be needed for concluding which system is better (Buckley and Voorhees, 2000).

The method works as follows: let *S* denote a set of runs. Let *x* and *y* denote a pair of runs from *S*. Let *Q* denote the entire evaluation collection. Let *f* represents the fuzziness value, which is the percent difference between scores such that if the difference is smaller than *f* then the two scores are deemed to be equivalent. We apply the algorithm of Figure 1 to obtain the information needed for computing the error rate (Formula (7)). *Stability* is inverse to this value, the lower the error rate is, the more stable the measure is. The same algorithm gives us the

proportion of ties (Formula (8)), which we use for measuring *discrimination power*, that is the lower the proportion of ties is, the more discriminative the measure is.

```
for each pair of runs x,y ϵ S
  for each trial from 1 to 100
    Q_i = select at random subcol of size c from Q;
    margin = f * max (M(x,Q_i),M(y,Q_i));
    if(|M(x,Q_i) - M(y,Q_i)| < |margin|)
      EQ_M(x,y)++;
    else if(|M(x,Q_i) > M(y,Q_i)|)
      GT_M(x,y)++;
    else
      GT_M(y,x)++;
```

Figure 1: Algorithm for computing $EQ_M(x,y)$, $GT_M(x,y)$ and $GT_M(y,x)$ in the stability method

We assume that for each measure the correct decision about whether run x is better than run y happens when there are more cases where the value of x is better than the value of y. Then, the number of times y is better than x is considered as the number of times the test is misleading, while the number of times the values of x and y are equivalent is considered the number of ties.

On the other hand, it is clear that larger fuzziness values decrease the error rate but also decrease the discrimination power of a measure. Since a fixed fuzziness value might imply different trade-offs for different metrics, we decided to vary the fuzziness value from 0.01 to 0.10 (following the work by Sakai (2007b)) and to draw for each measure a *proportion-of-ties / error-rate* curve. Figure 2 shows these curves for the *c@1*, *accuracy* and *UF* measures. In the Figure we can see how there is a consistent decrease of the error rate of all measures when the proportion of ties increases (this corresponds to the increase in the fuzziness value). Figure 2 shows that the curves of *accuracy* and *c@1* are quite similar (slightly better behavior of c@1) , which means that they have a similar stability and discrimination power.

The results suggest that the three measures are quite stable, having *c@1* and *accuracy* a lower error rate than *UF* when the proportion of ties grows. These curves are similar to the ones obtained for other QA evaluation measures (Sakai, 2007a).

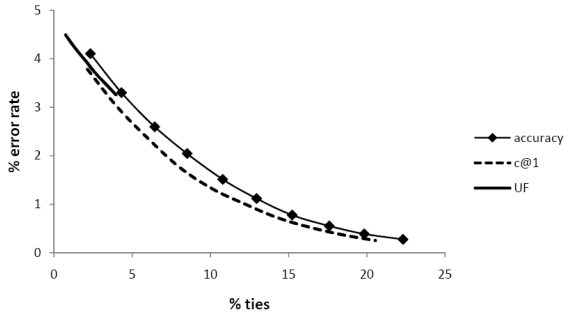

Figure 2: Error-rate / Proportion of ties curves for *accuracy*, *c@1* and *UF* with c = 250

4.3 Sensitivity

The *swap-rate* (Voorhees and Buckley, 2002) represents the chance of obtaining a discrepancy between two question sets (of the same size) as to whether a system is better than another given a certain difference bin. Looking at the swap-rates of all the difference performance bins, the performance difference required in order to conclude that a run is better than another for a given confidence value can be estimated. For example, if we want to know the required difference for concluding that system A is better than system B with a confidence of 95%, then we select the difference that represents the first bin where the swap-rate is lower or equal than 0.05.

The sensitivity of the measure is the number of times among all the comparisons in the experiment where this performance difference is obtained (Sakai, 2007b). That is, the more comparisons accomplish the estimated performance difference, the more *sensitive* is the measure. The more sensitive the measure, the more useful it is for system discrimination.

The swap method works as follows: let S denote a set of runs, let x and y denote a pair of runs from S. Let Q denote the entire evaluation collection. And let d denote a performance difference between two runs. Then, we first define 21 *performance difference bins*: the first bin represents performance differences between systems such that $0 \leq d < 0.01$; the second bin represents differences such that $0.01 \leq d < 0.02$; and the limits for the remaining bins increase by increments of 0.01, with the last bin containing all the differences equal or higher than 0.2.

$$Error\ rate_M = \frac{\sum_{x,y \epsilon S} min(GT_M(x,y), GT_M(y,x))}{\sum_{x,y \epsilon S}(GT_M(x,y) + GT_M(y,x) + EQ_M(x,y))} \quad (7)$$

$$Prop\ Ties_M = \frac{\sum_{x,y \epsilon S} EQ_M(x,y)}{\sum_{x,y \epsilon S}(GT_M(x,y) + GT_M(y,x) + EQ_M(x,y))} \quad (8)$$

Let BIN(d) denote a mapping from a difference *d* to one of the 21 bins where it belongs. Thus, algorithm in Figure 3 is applied for calculating the *swap-rate* of each bin.

```
for each pair of runs x,y ϵ S
  for each trial from 1 to 100
    select Q_i , Q_i' ⊂ Q, where
      Q_i ∩ Q_i' == φ and |Q_i| == |Q_i'| == c;
    d_M(Q_i) = M(x,Q_i) − M(y,Q_i);
    d_M(Q_i') = M(x,Q_i') − M(y,Q_i');
    counter(BIN(|d_M(Q_i)|))++;
    if(d_M(Q_i) * d_M(Q_i') < 0)
      swap_counter(BIN(|d_M(Q_i)|))++;
for each bin b
  swap_rate(b) = swap_counter(b)/counter(b);
```

Figure 3: Algorithm for computing swap-rates

	(i)	(ii)	(iii)	(iv)
UF	0.17	0.48	35.12%	59.30%
c@1	0.09	0.77	11.69%	58.40%
accuracy	0.09	0.68	13.24%	55.00%

Table 2: Results obtained applying the swap method to *accuracy*, *c@1* and *UF* at 95% of confidence, with c = 250: (i) Absolute difference required; (ii) Highest value obtained; (iii) Relative difference required ((i)/(ii)); (iv) percentage of comparisons that accomplish the required difference (sensitivity)

Given that Q_i and $Q_i^{'}$ must be disjoint, their size can only be up to half of the size of the original collection. Thus, we use the value c=250 for our experiment[1]. Table 2 shows the results obtained by applying the swap method to *accuracy*, *c@1* and *UF*, with c = 250, swap-rate ≤ 5, and sensitivity given a confidence of 95% (Column (iv)). The range of values are similar to the ones obtained for other measures according to (Sakai, 2007a).

According to Column (i), a higher absolute difference is required for concluding that a system is better than another using *UF*. However, the relative difference is similar to the one required by *c@1*. Thus, similar percentage of comparisons using *c@1* and *UF* accomplish the required difference (Column (iv)). These results show that their sensitivity values are similar, and higher than the value for *accuracy*.

4.4 Qualitative evaluation

In addition to the theoretical study, we undertook a study to interpret the results obtained by real systems in a real scenario. The aim is to compare the results of the proposed *c@1* measure with *accuracy* in order to compare their behavior. For this purpose we inspected the real systems runs in the data set.

System	c@1	accuracy	(i)	(ii)	(iii)
icia091ro	0.58	0.47	237	156	107
uaic092ro	0.47	0.47	236	264	0
loga092de	0.44	0.37	187	230	83
base092de	0.38	0.38	189	311	0

Table 3: Example of system results in QA@CLEF 2009. (i) number of questions correctly answered; (ii) number of questions incorrectly answered; (iii) number of unanswered questions.

Table 3 shows a couple of examples where two systems have answered correctly a similar number of questions. For example, this is the case of *icia091ro* and *uaic092ro* that, therefore, obtain almost the same *accuracy* value. However, *icia091ro* has returned less incorrect answers by not responding some questions. This is the kind of behavior we want to measure and reward. Table 3 shows how accuracy is sensitive only to the number of correct answers whereas *c@1* is able to distinguish when

[1]We use the same size for experiments in Section 4.2 for homogeneity reasons.

systems keep the number of correct answers but reduce the number of incorrect ones by not responding to some. The same reasoning is applicable to *loga092de* compared to *base092de* for German.

5 Related Work

The decision of leaving a query without response is related to the system ability to measure accurately its self-confidence about the correctness of their candidate answers. Although there have been one attempt to make the self-confidence score explicit and use it (Herrera et al., 2005), rankings are, usually, the implicit way to evaluate this self-confidence. Mean Reciprocal Rank (MRR) has traditionally been used to evaluate Question Answering systems when several answers per question were allowed and given in order (Fukumoto et al., 2002; Voorhees and Tice, 1999). However, as it occurs with *Accuracy* (proportion of questions correctly answered), the risk of giving a wrong answer is always preferred better than not responding.

The QA track at TREC 2001 was the first evaluation campaign in which systems were allowed to leave a question unanswered (Voorhees, 2001). The main evaluation measure was MRR, but performance was also measured by means of the percentage of answered questions and the portion of them that were correctly answered. However, no combination of these two values into a unique measure was proposed.

TREC 2002 discarded the idea of including unanswered questions in the evaluation. Only one answer by question was allowed and all answers had to be ranked according to the system's self-confidence in the correctness of the answer. Systems were evaluated by means of *Confidence Weighted Score* (*CWS*), rewarding those systems able to provide more correct answers at the top of the ranking (Voorhees, 2002). The formulation of CWS is the following:

$$CWS = \frac{1}{n}\sum_{i=1}^{n}\frac{C(i)}{i} \tag{9}$$

Where *n* is the number of questions, and *C(i)* is the number of correct answers up to the position *i* in the ranking. Formally:

$$C(i) = \sum_{j=1}^{i} I(j) \tag{10}$$

where *I(j)* is a function that returns 1 if answer *j* is correct and 0 if it is not. The formulation of *CWS* is inspired by the *Average Precision* (*AP*) over the ranking for one question:

$$AP = \frac{1}{R}\sum_{r} I(r)\frac{C(r)}{r} \tag{11}$$

where *R* is the number of known relevant results for a topic, and *r* is a position in the ranking. Since only one answer per question is requested, *R* equals to *n* (the number of questions) in *CWS*. However, in *AP* formula the summands belong to the positions of the ranking where there is a relevant result (product of I(r)), whereas in *CWS* every position of the ranking add value to the measure regardless of whether there is a relevant result or not in that position. Therefore, *CWS* gives much more value to some questions over others: questions whose answers are at the top of the ranking are giving almost the complete value to *CWS*, whereas those questions whose answers are at the bottom of the ranking are almost not counting in the evaluation.

Although *CWS* was aimed at promoting the development of better self-confidence scores, it was discussed as a measure for evaluating QA systems performance. *CWS* was discarded in the following campaigns of TREC in favor of *accuracy* (Voorhees, 2003). Subsequently, *accuracy* was adopted by the QA track at the Cross-Language Evaluation Forum from the beginning (Magnini et al., 2005).

There was an attempt to consider explicitly systems confidence self-score (Herrera et al., 2005): the use of the Pearson's correlation coefficient and the proposal of measures *K* and *K1* (see Formula 12). These measures are based in a utility function that returns -1 if the answer is incorrect and 1 if it is correct. This positive or negative value is weighted with the normalized confidence self-score given by the system to each answer. *K* is a variation of *K1* for being used in evaluations where more than an answer per question is allowed.

If the self-score is 0, then the answer is ignored and thus, this measure is permitting to leave a question unanswered. A system that always returns a

$$K1 = \frac{\sum_{i \epsilon \{correct_answers\}} self\_score(i) - \sum_{i \epsilon \{incorrect_answers\}} self\_score(i)}{n} \epsilon [-1, 1] \quad (12)$$

self-score equals to 0 (no answer) obtains a *K1* value of 0. However, the final value of *K1* is difficult to interpret: a positive value does not indicate necessarily more correct answers than incorrect ones, but that the sum of scores of correct answers is higher than the sum resulting from the scores of incorrect answers. This could explain the little success of this measure for evaluating QA systems in favor, again, of *accuracy* measure.

Accuracy is the simplest and most intuitive evaluation measure. At the same time is able to reward those systems showing good performance. However, together with MRR belongs to the set of measures that pushes in favor of giving always a response, even wrong, since there is no punishment for it. Thus, the development of better validation technologies (systems able to decide whether the candidate answers are correct or not) is not promoted, despite new QA architectures require them.

In effect, most QA systems during TREC and CLEF campaigns had an upper bound of accuracy around 60%. An explanation for this was the effect of error propagation in the most extended pipeline architecture: Passage Retrieval, Answer Extraction, Answer Ranking. Even with performances higher than 80% in each step, the overall performance drops dramatically just because of the product of partial performances. Thus, a way to break the pipeline architecture is the development of a module able to decide whether the QA system must continue or not its searching for new candidate answers: the Answer Validation module. This idea is behind the architecture of IBM's Watson (DeepQA project) that successfully participated at Jeopardy (Ferrucci et al., 2010).

In 2006, the first Answer Validation Exercise (AVE) proposed an evaluation task to advance the state of the art in Answer Validation technologies (Peñas et al., 2007). The starting point was the reformulation of Answer Validation as a Recognizing Textual Entailment problem, under the assumption that hypotheses can be automatically generated by combining the question with the candidate answer (Peñas et al., 2008a). Thus, validation was seen as a binary classification problem whose evaluation must deal with unbalanced collections (different proportion of positive and negative examples, correct and incorrect answers). For this reason, AVE 2006 used F-measure based on precision and recall for correct answers selection (Peñas et al., 2007). Other option is an evaluation based on the analysis of Receiver Operating Characteristic (ROC) space, sometimes preferred for classification tasks with unbalanced collections. A comparison of both approaches for Answer Validation evaluation is provided in (Rodrigo et al., 2011).

AVE 2007 changed its evaluation methodology with two objectives: the first one was to bring systems based on Textual Entailment to the Automatic Hypothesis Generation problem which is not part itself of the Recognising Textual Entailment (RTE) task but an Answer Validation need. The second one was an attempt to quantify the gain in QA performance when more sophisticated validation modules are introduced (Peñas et al., 2008b). With this aim, several measures were proposed to assess: the correct selection of candidate answers, the correct rejection of wrong answer and finally estimate the potential gain (in terms of accuracy) that Answer Validation modules can provide to QA (Rodrigo et al., 2008). The idea was to give value to the correctly rejected answers as if they could be correctly answered with the accuracy shown selecting the correct answers. This extension of accuracy in the Answer Validation scenario inspired the initial development of *c@1* considering non-response.

6 Conclusions

The central idea of this work is that not responding has more value than responding incorrectly. This idea is not new, but despite several attempts in TREC and CLEF there wasn't a commonly accepted mea-

sure to assess non-response. We have studied here an extension of *accuracy* measure with this feature, and with a very easy to understand rationale: Unanswered questions have the same value as if a proportion of them had been answered correctly, and the value they add is related to the performance (*accuracy*) observed over the answered questions. We have shown that no other estimation of this value produce a sensible measure.

We have shown also that the proposed measure *c@1* has a good balance of discrimination power, stability and sensitivity properties. Finally, we have shown how this measure rewards systems able to maintain the same number of correct answers and at the same time reduce the number of incorrect ones, by leaving some questions unanswered.

Among other tasks, measure *c@1* is well suited for evaluating Reading Comprehension tests, where multiple choices per question are given, but only one is correct. Non-response must be assessed if we want to measure effective reading and not just the ability to rank options. This is clearly not enough for the development of reading technologies.

Acknowledgments

This work has been partially supported by the Research Network MA2VICMR (S2009/TIC-1542) and Holopedia project (TIN2010-21128-C02).

References

Chris Buckley and Ellen M. Voorhees. 2000. Evaluating evaluation measure stability. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 33–40. ACM.

David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A. Kalyanpur, Adam Lally, J. William Murdock, Eric Nyberg, John Prager, Nico Schlaefer, and Chris Welty. 2010. Building Watson: An Overview of the DeepQA Project. *AI Magazine*, 31(3).

Junichi Fukumoto, Tsuneaki Kato, and Fumito Masui. 2002. Question and Answering Challenge (QAC-1): Question Answering Evaluation at NTCIR Workshop 3. In *Working Notes of the Third NTCIR Workshop Meeting Part IV: Question Answering Challenge (QAC-1), pages 1-10*.

Jesús Herrera, Anselmo Peñas, and Felisa Verdejo. 2005. Question Answering Pilot Task at CLEF 2004. In *Multilingual Information Access for Text, Speech and Images, CLEF 2004, Revised Selected Papers.*, volume 3491 of *Lecture Notes in Computer Science, Springer*, pages 581–590.

Bernardo Magnini, Alessandro Vallin, Christelle Ayache, Gregor Erbach, Anselmo Peñas, Maarten de Rijke, Paulo Rocha, Kiril Ivanov Simov, and Richard F. E. Sutcliffe. 2005. Overview of the CLEF 2004 Multilingual Question Answering Track. In *Multilingual Information Access for Text, Speech and Images, CLEF 2004, Revised Selected Papers.*, volume 3491 of *Lecture Notes in Computer Science, Springer*, pages 371–391.

Anselmo Peñas, Álvaro Rodrigo, Valentín Sama, and Felisa Verdejo. 2007. Overview of the Answer Validation Exercise 2006. In *Evaluation of Multilingual and Multi-modal Information Retrieval, CLEF 2006, Revised Selected Papers*, volume 4730 of *Lecture Notes in Computer Science, Springer*, pages 257–264.

Anselmo Peñas, Álvaro Rodrigo, Valentín Sama, and Felisa Verdejo. 2008a. Testing the Reasoning for Question Answering Validation. In *Journal of Logic and Computation. 18(3)*, pages 459–474.

Anselmo Peñas, Álvaro Rodrigo, and Felisa Verdejo. 2008b. Overview of the Answer Validation Exercise 2007. In *Advances in Multilingual and Multimodal Information Retrieval, CLEF 2007, Revised Selected Papers*, volume 5152 of *Lecture Notes in Computer Science, Springer*, pages 237–248.

Anselmo Peñas, Pamela Forner, Richard Sutcliffe, Álvaro Rodrigo, Corina Forascu, Iñaki Alegria, Danilo Giampiccolo, Nicolas Moreau, and Petya Osenova. 2010. Overview of ResPubliQA 2009: Question Answering Evaluation over European Legislation. In *Multilingual Information Access Evaluation I. Text Retrieval Experiments, CLEF 2009, Revised Selected Papers*, volume 6241 of *Lecture Notes in Computer Science, Springer*.

Alvaro Rodrigo, Anselmo Peñas, and Felisa Verdejo. 2008. Evaluating Answer Validation in Multi-stream Question Answering. In *Proceedings of the Second International Workshop on Evaluating Information Access (EVIA 2008)*.

Alvaro Rodrigo, Anselmo Peñas, and Felisa Verdejo. 2011. Evaluating Question Answering Validation as a classification problem. *Language Resources and Evaluation, Springer Netherlands (In Press)*.

Tetsuya Sakai. 2006. Evaluating Evaluation Metrics based on the Bootstrap. In *SIGIR 2006: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, Washington, USA, August 6-11, 2006*, pages 525–532.

Tetsuya Sakai. 2007a. On the Reliability of Factoid Question Answering Evaluation. *ACM Trans. Asian Lang. Inf. Process.*, 6(1).

Tetsuya Sakai. 2007b. On the reliability of information retrieval metrics based on graded relevance. *Inf. Process. Manage.*, 43(2):531–548.

Ellen M. Voorhees and Chris Buckley. 2002. The effect of Topic Set Size on Retrieval Experiment Error. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 316–323.

Ellen M. Voorhees and Dawn M. Tice. 1999. The TREC-8 Question Answering Track Evaluation. In *Text Retrieval Conference TREC-8*, pages 83–105.

Ellen M. Voorhees. 2001. Overview of the TREC 2001 Question Answering Track. In *E. M. voorhees, D. K. Harman, editors: Proceedings of the Tenth Text REtrieval Conference (TREC 2001). NIST Special Publication 500-250*.

Ellen M. Voorhees. 2002. Overview of TREC 2002 Question Answering Track. In *E.M. Voorhees, L. P. Buckland, editors: Proceedings of the Eleventh Text REtrieval Conference (TREC 2002). NIST Publication 500-251*.

Ellen M. Voorhees. 2003. Overview of the TREC 2003 Question Answering Track. In *Proceedings of the Twelfth Text REtrieval Conference (TREC 2003)*.