New Performance Metrics based on Multigrade Relevance: Their Application to Question Answering

Tetsuya Sakai
Knowledge Media Laboratory, Toshiba Corporate R&D Center
tetsuya.sakai@toshiba.co.jp

Abstract

This paper proposes two Information Retrieval performance metrics based on multigrade relevance, called Q-measure and R-measure, which are based on Cumulative Gain and Average Weighted Precision but are arguably more reliable. We also show how Q-measure and R-measure can be applied to Question Answering involving ranked lists of exact answers, and discuss the advantages of Q-measure over Reciprocal Rank through an experiment using the QAC1 test collection.

Keywords: *Q-measure, R-measure, Evaluation.*

1 Introduction

This paper proposes two Information Retrieval (IR) performance metrics based on multigrade relevance, called Q-measure and R-measure, which are based on Cumulative Gain [3] and Average Weighted Precision (originally called Weighted Average Precision [4]; See Section 2.3) but are arguably more reliable. We also show how Q-measure and R-measure can be applied to Question Answering (QA) evaluation involving ranked lists of exact answers, and discuss the advantages of Q-measure over Reciprocal Rank through an experiment using the QAC1 test collection. By providing full details of Q-measure and R-measure, including proofs, this paper serves as the backbone of our two NTCIR-4 site reports: Q-measure and R-measure are used as *IR* metrics with the NTCIR-4 CLIR test collections in [8], while Q-measure is used as a QA metric with the NTCIR-4 QAC2 test collection in [9].

In the early TREC English QA tracks (TREC-8 through TREC 2001) [10, 11], systems returned up to five candidate answers in decreasing order of confidence for the Main Task, i.e., "single-answer" task. Thus, if we let L and L' denote the system output size and the maximum output size allowed, respectively, then $L \leq L' = 5$ for all "single-answer" questions. *Reciprocal Rank* (RR) was used as the evaluation metric. TREC 2001 also introduced the List Task, in which systems were required to return an *unranked* list of answers. The answers were evaluated using *Accuracy*. The TREC List Task was *explicit* (up to TREC 2002) in that L' was clearly specified within each List question. However, these early TREC QA tracks dealt with fixed-length *text snippets* rather than exact answers.

The first Japanese Question Answering Challenge (QAC1) took place at NTCIR-3. QAC1 dealt with exact answers instead of text snippets, but basically followed the TREC QA evaluation methodology in that the Main Task (Task 1) used Reciprocal Rank with $L' = 5$. On the other hand, the QAC1 List Task (Task 2) used *F-measure* rather than accuracy for dealing with unranked answer lists, as the QAC1 List questions were in general *implicit*. Thus, in principle, the system had to determine the system output size L for each List question. (In fact, the QAC1 List question set was identical to the QAC1 Main question set, and the top performer in the List task simply let $L = 1$ for all questions.) The task settings for NTCIR-4 QAC2 are similar to those for QAC1.

Existing problems in QA evaluation include:

1. Different evaluation metrics need to be used for "different" QA tasks, as each of the metrics has its weaknesses: Reciprocal Rank can only look at the first correct response, while Accuracy and F-measure ignore answer priorities. However, the distinction between the above two tasks is not always clear, as there are more than one correct answer for many seemingly "single-answer" questions. Consider: Q: "What is the official language in Switzerland?" A: "Italian, German and French". It is also impossible to tell whether Q:"Who in Japan received the Nobel Prize in Physics?" is a list question or not unless you know the answer (or answers).

2. There is no QA evaluation metric that takes the *correctness level* of the answer into account. For example, for Q: "When did French revolutionaries storm the Bastille?" [10], A: "July 14, 1789" is more informative than A: "July 14" or

A: "1789". For Q: "Where is Tokyo Disneyland?" A:"Chiba prefecture" is probably more useful than A: "Japan". However, currently there is no way to reflect these differences.

Our new metrics, which are applicable to QA evaluation with ranked lists of exact answers, are designed to solve the above two problems. That is, we aim at integrating "single-answer" and "list" tasks as much as possible *and* incorporating answer correctness levels.

We are aware that Reciprocal Rank was abandoned at TREC 2002 with the requirement that the system must return *exactly one answer* (i.e. $L = L' = 1$) for the Main Task, and that CLEF 2004 is also following this move. However, we believe that evaluating ranked lists for QA is still important for the following reasons:

1. Returning a single exact answer is not the only possibility in practical QA systems. That is, a small ranked list of possible answers may be perfectly acceptable for some applications, e.g., when *answer recall* is considered to be important.

2. From a statistical viewpoint, evaluation based on single answers may not be reliable, as this is like measuring document retrieval performance by examining the document at Rank 1 only. Thus, a very good system that unluckily returned a correct answer at Rank 2 for *all* questions would be judged as "complete rubbish". To circumvent this danger, a large question set is often used, which can be burdensome for test collection constructors.

3. There appears to be some room for improvement in QA evaluation with $L' = 1$. The aim of introducing *Confidence Weighted Score* (CWS) at TREC 2002 was to measure a system's ability to recognise when it has found a correct answer to a given question [12]. However, it is clear from its definition that CWS only measures the system's ability to determine whether it is more confident about one question than another in a given question set: that is, it only measures *relative* confidence. Moreover, the idea of ranking questions in CWS may be counter-intuitive in some cases: for example, two TREC 2002 systems had nearly identical CWS values even though one system answered 28 more questions than the other one [12].

The remainder of this paper is organised as follows. Section 2 re-examines existing IR metrics, and Section 3 proposes new IR metrics for multigrade relevance called Q-measure and R-measure, as well as how to apply them to QA evaluation. Section 4 describes an experiment using the QAC1 QA test collection to discuss the advantages of Q-measure over Reciprocal Rank. Section 5 discusses extensions and limitations of the present work, and Section 6 concludes this paper.

2 Existing Metrics

2.1 Average Precision

Average Precision (e.g. [2]) is one of the most widely-used IR metric, although it cannot handle multigrade relevance. Let R denote the total number of known relevant documents for a particular search request (or a *topic*), and let $count(r)$ denote the number of relevant documents within the top r documents of the ranked output. Clearly, the Precision at Rank r is $count(r)/r$. Let $isrel(r)$ denote a binary flag, such that $isrel(r) = 1$ if the document at Rank r is relevant and $isrel(r) = 0$ otherwise. Then, Average Precision (AP) is defined as:

$$AP = \frac{1}{R} \sum_{1 \leq r \leq L} isrel(r) \frac{count(r)}{r} \qquad (1)$$

where L is the ranked output size.

Another useful measure often used along with AP is *R-Precision*:

$$R\text{-}Precision = \frac{count(R)}{R} \qquad (2)$$

These measures are known to "average well" across a set of topics, in contrast to metrics that are based on *fixed* document ranks (See Section 2.2).

2.2 Cumulative Gain

Järvelin and Kekäläinen proposed (Discounted) Cumulative Gain for evaluation based on multigrade relevance [3]. Their basic idea is that a system output, scanned from the top, receives a score for each retrieved relevant document. The score for retrieving a highly relevant document is high, and that for retrieving a partially relevant one is low.

Formally, let X denote a relevance level, and let $gain(X)$ denote the *gain value* for successfully retrieving an X-relevant document. For the NTCIR CLIR test collections, $X \in \{S, A, B\}$ [4], and a typical gain value assignment would be $gain(S) = 3, gain(A) = 2, gain(B) = 1$. Hereafter, we use the above NTCIR relevance levels and the gain value assignment without loss of generality. Let $X(r)$ denote the relevance level of the document at Rank r ($\leq L$). Then, the *gain at Rank* r is given by $g(r) = gain(X(r))$ if the document at Rank r is relevant, and $g(r) = 0$ if it is nonrelevant. The *cumulative gain at Rank* r is given by $cg(r) = g(r) + cg(r-1)$ for $r > 1$ and $cg(1) = g(1)$.

Järvelin and Kekäläinen used the Cumulative Gain by averaging $cg(r)$ across a given topic set for each r, from the viewpoint of *how many documents the user has to go through*. However, as Kando *et al.* [4] and Sakai [6] have pointed out, this is not desirable

from a *statistical* viewpoint, as the number of relevant documents (R) differs across the search request set, and therefore the upperbound performance at a fixed rank differs across the set as well. (This also applies to Precision at a fixed document rank.) For example, consider a ranked output with three nonrelevant documents and two B-relevant documents at the very top, such that its *gain sequence* is $(g(1), g(2), \ldots) = (0, 0, 0, 1, 1)$, so that its *cumulative gain sequence* is $(cg(1), cg(2), \ldots) = (0, 0, 0, 1, 2)$. Let $R(X)$ denote the number of known X-relevant documents so that $\sum_X R(X) = R$, and suppose that such a ranked output was returned for both Topic One with $R = R(B) = 2$, and for Topic Two with $R = R(B) = 100$. Then, for *both* of these topics, the Precision at Rank 5 is 2/5=0.4 and the Cumulative Gain at Rank 5 is $cg(5) = 2$. However, these values clearly represent the *best possible* performance at Rank 5 for Topic One, while they are far from it for Topic Two.

2.3 Average Weighted Precision

Average Weighted Precision (AWP) proposed by Kando *et al.* [4] is based on Cumulative Gain, but is arguably more statistically reliable as it performs comparison with an *ideal* (i.e. best possible [3]) ranked output before averaging across topics. Let $cig(r)$ represent the cumulative gain at Rank r for an ideal ranked output. (An ideal ranked output for NTCIR can be obtained by listing up all S-relevant documents, then all A-relevant documents, then all B-relevant documents.) Then, AWP is given by:

$$AWP = \frac{1}{R} \sum_{1 \le r \le L, g(r) > 0} \frac{cg(r)}{cig(r)}$$

$$= \frac{1}{R} \sum_{1 \le r \le L} isrel(r) \frac{cg(r)}{cig(r)} \quad (3)$$

For binary relevance such that each relevant document gives a gain value of 1, AWP becomes very similar to Average Precision, as $cg(r)/cig(r) = count(r)/r$ for $r \le R$. Thus $cg(r)/cig(r)$ is a kind of "weighted precision" based on relevance levels. (Hence, from Equation 3, we prefer the name Average Weighted Precision to Weighted Average Precision.)

Similarly, Kando *et al.* have proposed an extension of R-precision, called *R-Weighted Precision* (originally called Weighted R-Precision):

$$R\text{-}WP = \frac{cg(R)}{cig(R)} \quad (4)$$

3 Proposed Metrics

3.1 Q-measure and R-measure

Although AWP appears to be a natural extension of Average Precision for dealing with multigrade relevance, it suffers from a problem. Consider an extreme case in which there is only one known relevant document, which is B-relevant (i.e. $R = R(B) = 1$). As the ideal ranked output for this case should have the B-relevant document at Rank 1, the sequence of $cig(r)$ is clearly $(1, 1, 1, \ldots)$. Thus, for System A which returned a relevant document at Rank 1, its sequence of $cg(r)$ is also $(1, 1, 1, \ldots)$, and therefore its AWP is $(cg(1)/cig(1))/1 = (1/1)/1 = 1$. Now, suppose that Sytem B returned 99 nonrelevant documents before returning the relevant one at Rank 100. Clearly, its $cg(r)$ is 0 for $1 \le r \le 99$, and 1 for $r \ge 100$. Surprisingly, its AWP is also $(cg(100)/cig(100))/1 = (1/1)/1 = 1$. That is, System B is regarded as identical in performance to System A. This problem arises from the fact that $cig(r)$ does not increase with r after it has run out of relevant documents, i.e. after Rank R. That is, while it is guaranteed that $cig(r) > cig(r-1)$ for $r \le R$, unfortunately $cig(r) = cig(r-1)$ holds for $r > R$. This means that, after Rank R, *AWP cannot impose a penalty for going down the ranked list*. Average Precision is free from this problem, because it uses the actual Rank r instead of $cig(r)$ as the denominator, which is guaranteed to increase steadily. (Compare Equations (1) and (3)). R-WP and R-precision are also free from this problem because they ignore Ranks after R.

We now propose Q-measure and R-measure to solve the above problem. First, we introduce the notion of *bonused gain at Rank* r, simply given by $bg(r) = g(r) + 1$ if $g(r) > 0$ and $bg(r) = 0$ if $g(r) = 0$. Then, the *cumulative bonused gain at Rank* r is given by $cbg(r) = bg(r) + cbg(r-1)$ for $r > 1$ and $cbg(1) = bg(1)$. Q-measure is defined as:

$$Q\text{-}measure = \frac{1}{R} \sum_{1 \le r \le L, g(r) > 0} \frac{cbg(r)}{cig(r) + r}$$

$$= \frac{1}{R} \sum_{1 \le r \le L} isrel(r) \frac{cbg(r)}{cig(r) + r} \quad (5)$$

The denominator in the above equation ($cig(r) + r$) increases by at least r as we go down the ranked list, *even after Rank* R. In contrast, the numerator ($cbg(r)$) receives a bonus point of one only if the document at Rank r is relevant. Thus, Q-measure is equal to one if and only if the system output is an ideal one, provided that $L \ge R$: the proof is given in the **Appendix**. Recall the case with only one B-relevant document mentioned earlier in this section: The sequence of $cbg(r)$ for System A, which returned the B-relevant document at the top, is $(\mathbf{2}, 2, 2, \ldots)$, and therefore $Q\text{-}measure = (cbg(1)/(cig(1) + 1))/1 = (\mathbf{2}/(\mathbf{1} + 1))/1 = 1$. (Hereafter, $cbg(r)$ and $cig(r)$ are shown in **bold** whenever $g(r) > 0$.) On the other hand, for System B, which returned the B-relevant document at Rank 100, $cbg(r) = 0$ for $1 \le r \le 99$ and $cbg(100) =$

2. Thus, $Q\text{-}measure = (cbg(100)/(cig(100) + 1))/1 = (\mathbf{2}/(\mathbf{1} + 100))/1 = 0.0198$.

As mentioned earlier, R-Precision and R-WP do not share the aforementioned problem with AWP. However, just as R-precision is used besides Average Precision, it is possible to devise a counterpart of Q-measure by analogy:

$$R\text{-}measure = \frac{cbg(R)}{cig(R) + R} \quad (6)$$

Again, R-measure is equal to one if and only if the system output is an ideal one. For example, for the aforementioned System A, $R\text{-}measure = cbg(1)/(cig(1) + 1) = \mathbf{2}/(\mathbf{1} + 1) = 1$, while, for System B, $R\text{-}measure = cbg(1)/(cig(1) + 1) = 0/(1 + 1) = 0$. As this example suggests, R-measure is a demanding metric if R is small. (This applies to R-WP and R-Precision as well.)

3.2 Application to Question Answering

This section describes how to apply Q-measure (and R-measure) to QA evaluation involving ranked lists of exact answers. The difficuly of QA evaluation lies in the fact that arbitrary answer strings need to be evaluated, in contrast to an IR situation in which only a closed-class, unique document IDs need to be evaluated. To overcome this problem (at least partially; See Section 5.2), we propose to provide equivalence classes of answers, or *answer synsets*, at the time of QA test collection construction. Using answer synsets, we can handle both "single-answer" and "list" questions in an "answer ranking" task, and can avoid rewarding systems that return duplicate answers that mean the same thing. Of course, the concept of answer synset itself is not new, as the List Tasks at TREC and NTCIR have evaluated system output based on the number of *distinct* correct answers. What is new is that we propose to assign a *correctness level* to each answer string within each answer synset.

Let $AS(i)$ $(1 \leq i \leq R)$ denote an answer synset, and let $a(i, j)$ denote the j-th answer string in $AS(i)$. Let $x(i, j)$ denote the correctness level of $a(i, j)$, and let $xmax(i) = max_j x(i, j)$. That is, $xmax(i)$ is the *highest correctness level* within $AS(i)$. Then we define $R(X)$ as the number of answer synsets such that $xmax(i) = X$. Thus, if we extend the NTCIR document relevance levels to answer correctness levels, $R(S) + R(A) + R(B) = R$.

Below, we show some examples of how to prepare QA test collections in this way.

Example 1:
Q: "Who played in the Beatles?" ($R = R(S) = 4$)
$AS(1)$ = {<"Sir Paul McCartney",S >, <"Paul McCartney",S >, <"McCartney",A >, <"Paul",B >}
$AS(2)$ = {<"John Lennon",S >, <"Lennon",A >, <"John",B >}
$AS(3)$ = {<"George Harrison",S >, <"Harrison",A >, <"George",B>}
$AS(4)$ = {<"Ringo Starr",S >, <"Starr",A >, <"Ringo",B >}
Some test collection constructors may prefer to add more answer synsets with relatively low correctness levels, representing early/temporary members of the Beatles, such as:
$AS(5)$ = {<"Stuart Sutcliffe",B >, <"Sutcliffe",B >, <"Stuart",B >}
If the fifth answer synset is added, then $R(B) = 1$ and therefore $R = R(S) + R(B) = 5$.

Example 2:
Q: "What does DVD stand for?" ($R = R(S) = 1$)
$AS(1)$ = {<"Digital Versatile Disk",S >,<"Digital Video Disk",A >}
If a system that returns *both* of the above answer strings is preferrable, then the above data should be broken into two separate answer synsets.

Example 3:
Q: "What is love?" ($R = R(A) = 1$)
$AS(1)$ = {<"NIL",A >}
The answer data for NIL questions should be prepared as above. The correctness level of the NIL answer does not affect the QA performance, as we shall see later.

Figures 1 and 2 show an example of how to implement Q-measure and R-measure calculation for QA. Firstly, the algorithm in Figure 1 reads a ranked list of answers and marks the correct ones with S, A or B, but avoids marking duplicate answers from the same answer synset. Then, the algorithm in Figure 2 reads the above *marked* answers to calculate Q-measure and R-measure. (Figure 1 includes a special treatment of NIL answers: only a NIL answer at Rank 1 is marked as correct, in contrast to the TREC 2001 evaluation in which systems could be rewarded for including "NIL" somewhere in the ranked list [11].)

Let us return to *Example 1* (without the fifth answer synset), and suppose that the system output was ("McCartney", "Lennon", "Paul", "George Harrison", "Starr"). Then, $(g(1), g(2), \ldots) = (2, 2, 0, 3, 2)$, and $(bg(1), bg(2), \ldots) = (3, 3, 0, 4, 3)$. Hence $(cbg(1), cbg(2), \ldots) = (\mathbf{3}, \mathbf{6}, 6, \mathbf{10}, \mathbf{13})$. Whereas, an example ideal ranked output for this question is ("Paul McCartney", "John Lennon", "George Harrison", "Ringo Starr"), so that $(cig(1), cig(2), \ldots) = (\mathbf{3}, \mathbf{6}, 9, \mathbf{12}, \mathbf{12}, \ldots)$. Therefore, $Q\text{-}measure = (\mathbf{3}/(\mathbf{3} + 1) + \mathbf{6}/(\mathbf{6} + 2) + \mathbf{10}/(\mathbf{12} + 4) + \mathbf{13}/(\mathbf{12} + 5))/4 = 0.722$, and $R\text{-}measure = \mathbf{10}/(\mathbf{12} + 4) = 0.625$.

For *Example 3* (where "NIL" is regarded as A-correct), if the system correctly returns "NIL" at Rank 1, then $(g(1), g(2), \ldots) = (2, 0, \ldots)$, $(bg(1), bg(2), \ldots) = (3, 0, \ldots)$, and $(cbg(1), cbg(2), \ldots) = (\mathbf{3}, 3, \ldots)$. Whereas, $(cig(1), cig(2), \ldots) = (\mathbf{2}, 2, \ldots)$. Thus,

```
/* initialize flag for each answer synset.
The flags avoid marking multiple answers from
the same answer synset. */
for( i=1; i<=R; i++ ) flag[i]=0;

r=1; /* system output rank */
while read o(r){ /* system's r-th answer */
  if( there exists a(i,j) s.t. o(r)==a(i,j) ){
  /* o(r) matches with a correct answer */
    if( o(r)=="NIL" ){
    /* special treatment of NIL */
      if( r==1 ){ /* i.e. NIL at Rank 1 */
        print o(r), x(i,j);
        /* marked as correct */
      }
      else{
        print o(r);
        /* NOT marked as correct */
      }
    }
    else{ /* not NIL */
      if( flag[i]==0 ){
      /* AS(i) is a NEW answer synset */
        print o(r), x(i,j);
        /* marked as correct */
        flag[i]=1;
      }
      else{ /* i.e. flag[i]==1 */
        print o(r);
        /* duplicate answer from AS(i)
        NOT marked as correct */
      }
    }
  }
  else{ /* no match with a correct answer */
    print o(r);
    /* NOT marked as correct */
  }
  r++; /* examine next rank */
}
```

Figure 1. Algorithm for marking a system output.

```
rmax=max(L,R); /* L: system output size */
               /* R: #answer synsets */

/* obtain cumulative gains for the
IDEAL ranked output */
r=0; cig[0]=0;
for each X in (S,A,B) { /* X: correctness level */
  for( k=1; k<=R(X); k++ ){
  /* R(X): #answer synsets in which the
     highest correctness level is X. */
    r++;
    cig[r]=cig[r-1]+gain(X);
  }
}
for( r=R+1; r<=rmax; r++ ){ /* in case L>R */
  cig[r]=cig[R];
}

/* obtain cumulative bonused gains for
the system output */
r=0; cbg[0]=0;
for( r=1; r<=L; r++ ){
  if( o(r) is marked with X ){
    cbg[r]=cbg[r-1]+gain(X)+1;
  }
  else{
    cbg[r]=cbg[r-1];
  }
}
for( r=L+1; r<=rmax; r++ ){ /* in case L<R */
  cbg[r]=cbg[L];
}

/* calculation */
sum=0;
for( r=1; r<=L; r++ ){
  if( cbg[r]>cbg[r-1] ){
  /* i.e. correct answer at Rank r */
    sum+=cbg[r]/(cig[r]+r);
  }
}
Q-measure=sum/R;
R-measure=cbg[R]/(cig[R]+R);
```

Figure 2. Algorithm for calculating Q-measure/R-measure.

$Q\text{-}measure = R\text{-}measure = \mathbf{3}/(\mathbf{2}+1) = 1$. In general, if the answer at Rank 1 is correct, $cbg(1) = bg(1) = g(1) + 1$ and $cig(1) = g(1)$ hold, hence $cbg(1)/(cig(1)+1) = 1$. Therefore, the NIL answer at Rank 1 would receive a Q/R-measure of 1.0 whether it is treated as S-, A- or B-correct.

4 Experiments

This section discusses the advantages of Q-measure over Reciprocal Rank through an experiment using the QAC1 test collection.

4.1 Extended QAC1 Collection

To use Q-measure and R-measure with the QAC1 test colleciton, the author manually converted the "flat" answer data of QAC1 into answer synsets, and assigned a correctness level to each answer string. Although we could not hire a second judge for enhancing the reliability of the new answer data, here we assume that inter-judge differences do not affect comparative evaluation [10]. (Strictly speaking, however, whether inter-judge differences in defining answer synsets and multigrade relevance affect evaluation is an open question. More importantly, the *reusability* of the QAC1 test collection has never been guaranteed: it is known that QA test collections are inherently less reusable than IR test collections [10].)

We were able to add answer synsets and correctness levels to the original QAC1 answer data without any major problems. (We have also constructed our own QA test collections, and from our experience, it is not so difficult to find answer strings, define answer synsets, and assign correctness levels at the same time.) Table 1 (a) shows the distribution of the number of answer synsets for the Extended QAC1 data: it can be observed that there is only one answer synset (i.e. $R = 1$) for 161 questions. Thus, R-measure may be too demanding for this test collection as it only evaluates top R answers. (Note also that Kando's AWP is clearly not suitable for QA evaluation: As have been discussed in Section 3.1, if $R = 1$, then $cig(r)$ remains constant for all r. Therefore, from Equation 3, System A that returns the correct answer at Rank 1 and System B that returns the same answer at Rank 5

would receive the same score.)

The outlier with $R = 18$ in Table 1 (a) is a very ambiguous List question: QAC1-1097 ("What are the Three Sacred Treasures?"). Although the phrase "Three Sacred Treasures" originally refer to specific historic items that symbolise the Imperial Throne, it is often used in newspaper contexts such as "Three Sacred Treasures of the Modern Era". Thus, consumer products such as "color TV" and "refrigerator" are included in the original answer set. Ideally, such outlier questions should be discarded from the evaluation set, because, if the system output size L is smaller than R, it is impossible to achieve a Q-measure of 1.

Table 1 (b) shows the distribution of correctness levels of the QAC1 answer strings. As there are 282 answer *synsets* in total, each answer synset contains 616/282=2.18 answer strings on average.

As *supporting documents* [10, 11, 12] were not evaluated at QAC1, our Extended QAC1 data are based on answer strings rather than answer-document pairs. Thus our evaluation is *lenient* in TREC parlance.

Table 1. Distribution of R and correctness levels for the 195 QAC1 questions.

(a)	
R	#questions
1	161
2	14
3	12
4	4
5	1
9	2
18	1
total	195

(b)	
correctness level	#answer strings
S	401
A	118
B	97
total	616

Table 2. Performance of ASKMi for the 195 QAC1 questions.

	RR	Q-measure	R-measure
AF	0.682	0.684	0.543
	4↓26↑	5↓35↑	4↓20↑
noAF	0.637	0.639	0.469

4.2 ASKMi Japanese QA System

ASKMi, the Japanese QA system used in the present experiments, is described fully in [7, 9], and it suffices to treat it as a "black box" for the purpose of this study. To illustrate the advantages of Q-measure over Reciprocal Rank, this paper examines two ASKMi runs, namely, those with and without the *Answer Formulator* module. The primary function of the Answer Formulator is *answer string consolidation*: For example, if the original ranked list of answers contains "*Koizumi shushō* (prime minister Koizumi)" at Rank 1 and "*Koizumi*" at Rank 4, the answer formulator tries to erase the latter to minimise redundancy.

4.3 Results and Discussions

Table 2 summarises the performance of ASKMi for the 195 non-NIL questions from QAC1. (Currently, ASKMi cannot detect NIL questions.) The runs with and without the Answer Formulator are represented by **AF** and **noAF**, respectively. The table also shows question-by-question comparisons: for example, in terms of Reciprocal Rank, **AF** outperforms **noAF** for 26 questions while **noAF** outperforms **AF** for 4 questions. While these differences are statistically significant with the Sign Test for all three metrics, it can be observed that Q-measure is *more sensitive* than Reciprocal Rank: while Q-measure detected a performance difference for 40 questions (5 down and 35 up), Reciprocal Rank detected a performance difference for only 30 questions (4 down and 26 up). R-measure, on the other hand, appears to be less sensitive to the effect of Answer Formulator, as R is generally very small for the QAC1 questions.

Although the *Mean* Reciprocal Rank and the *Mean* Q-measure values are very similar for this test collection, individual values are in fact quite different. Among the 195 questions, there were 23 questions for which the **AF** performance was 1.0 in terms of Reciprocal Rank *and* less than one in terms of Q-measure. This happens when a system returns a (somewhat) correct answer at Rank 1 *and*: (a) the above answer is not the *best* answer; or (b) there is at least one more answer synset and the system did not handle it well. An example of (a) is QAC1-1012 "When did Yasunari Kawabata become the first Japanese to receive the Nobel Prize in Literature?". **AF**'s first response for this question was "1968", which was only B-relevant. Thus, $cbg(1) = bg(1) = g(1) + 1 = \mathbf{2}$. There was only one answer synset for this question, which included "December 10, 1968" as an S-correct answer. Thus, $R = R(S) = 1$, and $cig(1) = \mathbf{3}$. Therefore, $Q\text{-}measure = (\mathbf{2}/(\mathbf{3}+1))/1 = 0.5$. An example of (b) is QAC1-1058 "Japanese who received the Nobel Prize in Physics". The **AF** run returned "Hideki Yukawa" at Rank 1 and "Shinichiro Tomonaga" at Rank 5, both of which are S-correct. Thus, the bonused gain sequence is $(4, 0, 0, 0, 4)$ and the cumulative bonused gain sequence is $(\mathbf{4}, 4, 4, 4, \mathbf{8})$. There are three answer synsets (representing three researchers) and $R = R(S) = 3$ for this question. Thus, $(cig(1), cig(2), cig(3), \ldots) = (\mathbf{3}, 6, 9, 9, \mathbf{9}, \ldots)$. Therefore, $Q\text{-}measure = (\mathbf{4}/(\mathbf{3}+1)+\mathbf{8}/(\mathbf{9}+5))/3 = 0.524$. Note that Reciprocal Rank ignores the correct answer at Rank 5 completely, and would have fully accepted "incomplete" answers such as "Yukawa".

Figure 3 visualises the performance differences be-

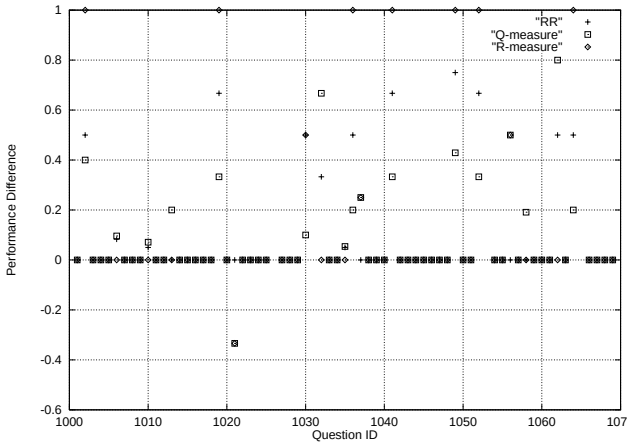

Figure 3. Performance difference (AF-noAF) for QAC1 (1001-1069).

tween **AF** and **noAF** in terms of each evaluation metric for the first one-third of the QAC1 questions. Thus, dots above and below zero represent the positive and negative effects of the Answer Formulator, respectively, and they correspond to the "arrows" in Table 2. Although the Answer Formulator can occasionally hurt performance, some of the seemingly negative effects are because of the aforementioned *reusability* problem. For example, the only "negative dot" in Figure 3 represents QAC1-1021 "How was Prime Minister Obuchi criticized just after inauguration?": The **noAF** run returned "ordinary man" at Rank 1 and "cold pizza" at Rank 2, both of which were S-correct. However, the Answer Formulator replaced "cold pizza" with "Obuchi is as uninspiring as cold pizza", as it judged the longer answer to be more informative. Unfortunately, the longer answer string was beyond the scope of QAC1 and was *not* listed as a correct answer. Hence the **AF** run received a lower score.

Let us go back to the discussion of the *sensitiveness* of QA metrics in terms of comparison between **AF** and **noAF**. For QAC1-1013, 1021, 1037, 1056 and 1058 in Figure 3, the difference in terms of Reciprocal Rank is zero while that in terms of Q-measure is not. That is, for these questions, Q-measure detected the effect of the Answer Formulator which Reciprocal Rank overlooked. For QAC1-1058 mentioned earlier in this section, the **noAF** run failed to return the second correct answer "Shinichiro Tomonaga", as its answer list contained *duplicates*, namely, "Hideki Yukawa" at Rank 1 and "Doctor Hideki Yukawa" at Rank 4. Thus, after answer string consolidation, "Shinichiro Tomonaga" rose to Rank 5 and received credit in terms of Q-measure. Also, we have examined QAC1-1021 already. These examples show that Q-measure not only handles both "single-answer" and "list" questions seamlessly but also evaluates the system's power to minimise redundancy.

R-measure is also more sensitive than Reciprocal Rank for QAC1-1021, 1037 and 1056 in Figure 3: for these questions, the R-measure values were actually equal to the Q-measure ones. However, as mentioned earlier, R-measure can be insensitive to changes in the ranked list for questions with small R. For example, for QAC1-1006 "When will NTT Communications take over NTT International Network?" included in Figure 3, the difference in R-measure is zero while those in Reciprocal Rank and Q-measure are 0.083 and 0.096, respectively. Although the Answer Formulator managed to move the correct answer "October 1" from Rank 4 to Rank 3 by erasing "October" (treated as incorrect in the QAC1 data) which was originally at Rank 3, R-measure did not detect this improvement because, for this question, $R = R(S) = 1$. Probably, R-measure is more suitable for IR than for QA.

5 Extensions and Limitations

5.1 IR evaluation and Average Gain Ratio

Recently, Sakai [6] has proposed *Average Gain Ratio* (AGR) and *R-Gain Ratio* for IR evaluation based on multigrade relevance. These metrics are the same as Kando's AWP and R-WP, respectively, except that they use *topic adjusted* gain values instead of *fixed* gain values such as $gain(S) = 3, gain(A) = 2, gain(B) = 1$. Thus, Sakai proposes to perform the following transformation for each topic:

$$gain'(X) = gain(X) - \frac{R(X)}{R}(gain(X) - gain(X')) \quad (7)$$

where X' is the relevance level that is one level lower than X. (If X is the lowest relevance level, then $gain(X')$ is taken to be zero. Moreover, the above transformation is not applied if $R(X) = R$.) The above transformation was proposed based on the observation that the ratio $R(S) : R(A) : R(B)$ differs considerably across topics for the NTCIR CLIR test collections. For example, $R(B) >> R(S)$ for many questions, but not for *all* questions.

Although AGR inherits the problem of AWP discussed in Section 3.1, Equation 7 can easily be applied to Q-measure and R-measure as well.

5.2 Definition/Why/How Questions

Clearly, our evaluation methodology cannot fully handle definition/why/how type questions as it is almost impossible to prepare *exhaustive* lists of such answers in advance. Although some automatic evaluation methods based on comparison with gold-standard texts have been proposed for Machine Translation, Summarisation and QA [1, 5], problems remain for

QA: Suppose that the user asks "What is exothermic reaction?" and the system responds with "a chemical reaction accompanied by the absorption of heat". The correct answer is, however, "a chemical reaction accompanied by the *evolution* of heat". Using existing automatic evaluation metrics, the system would receive a high score despite the fact that it is telling a complete lie, as the two answer strings do share word N-grams and are identical in length [5].

6 Conclusions

We have proposed Q-measure and R-measure, which are statistically reliable IR metrics for multi-grade relevance. Through an experiment using the QAC1 test collection, we also showed that Q-measure can handle both "single-answer" and "list" questions in QA evaluation with ranked lists of exact answers.

Appendix: Proof that Q-measure is equal to one iff the system output is an ideal one (provided that $L \geq R$).

Given that the system output is an ideal one, then $cg(r) = cig(r)$ holds for $r \geq 1$. Moreover, $bg(r) = g(r)+1$ holds for $r \leq R$ as all of the top R documents should be relevant. Therefore, for $r \leq R$,

$$cbg(r) = \sum_{1 \leq r' \leq r} bg(r') = \sum_{1 \leq r' \leq r} (g(r') + 1)$$

$$= cg(r) + r = cig(r) + r \qquad (8)$$

Now, since the system output is an ideal one, there should be no relevant document below Top R, thus $g(r) = 0$ holds for $r > R$. Therefore,

$$Q\text{-}measure = \frac{1}{R} \sum_{r,g(r)>0} \frac{cbg(r)}{cig(r) + r}$$

$$= \frac{1}{R} \sum_{1 \leq r \leq R} \frac{cbg(r)}{cig(r) + r} = \frac{1}{R} \sum_{1 \leq r \leq R} \frac{cig(r) + r}{cig(r) + r} = 1.$$

Conversely, given that Q-measure is 1, then

$$R = \sum_{r,g(r)>0} \frac{cbg(r)}{cig(r) + r} \qquad (9)$$

holds. Now, since $bg(r)$ receives a bonus point of 1 (i.e. $bg(r) = g(r) + 1$) only when the document at Rank r is relevant, $cbg(r) \leq cg(r) + r$ holds for $r \geq 1$. Moreover, by definition of $cig(r)$, $cg(r) \leq cig(r)$ holds for $r \geq 1$. Therefore,

$$\frac{cbg(r)}{cig(r) + r} \leq \frac{cg(r) + r}{cig(r) + r} \leq 1 \qquad (10)$$

holds for $r \geq 1$. From Equations (9) and (10),

$$R \leq \sum_{r,g(r)>0} \frac{cg(r) + r}{cig(r) + r} \leq \sum_{r,g(r)>0} \qquad (11)$$

holds. However, as there are no more than R relevant documents, $R \geq \sum_{r,g(r)>0}$ should hold for $r > 1$. From this and Equation (11), both

$$R = \sum_{r,g(r)>0} \qquad (12)$$

and

$$\sum_{r,g(r)>0} \frac{cg(r) + r}{cig(r) + r} = \sum_{r,g(r)>0} \qquad (13)$$

hold for $r > 1$. Equation (12) implies that the system output includes *all* relevant documents. Meanwhile, From Equations (10) and (13), it is necessary that $cg(r) = cig(r)$ for every r s.t. $g(r) > 0$. Therefore, the system output must be an ideal one.

References

[1] Breck, E. J. *et al.*: How to Evaluate Your Question Answering System Every Day... and Still Get Real Work Done, *LREC 2000 Proceedings*, 2000.

[2] Buckley, C. and Voorhees, E. M.: Evaluating Evaluation Measure Stability, *ACM SIGIR 2000 Proceedings*, pp. 33-40, 2000.

[3] Järvelin, K. and Kekäläinen, J.: IR Evaluation Methods for Retrieving Highly Relevant Documents, *ACM SIGIR 2000 Proceedings*, pp. 41-48, 2000.

[4] Kando, N., Kuriyama, K. and Yoshioka, M.: Information Retrieval System Evaluation using Multi-Grade Relevance Judgments - Discussion on Averageable Single-Numbered Measures (in Japanese), *IPSJ SIG Notes*, FI–63–12, pp. 105-112, 2001.

[5] Lin, C.-Y. and Hovy, E.: Automatic Evaluation of Summaries Using N-gram Co-Occurrence Statistics, *HLT-NAACL 2003 Proceedings*, 2003.

[6] Sakai, T.: Average Gain Ratio: A Simple Retrieval Performance Measure for Evaluation with Multiple Relevance Levels, *ACM SIGIR 2003 Proceedings*, pp. 417–418, 2003.

[7] Sakai, T. *et al.*: ASKMi: A Japanese Question Answering System based on Semantic Role Analysis, *RIAO 2004 Proceedings*, pp. 215–231, 2004.

[8] Sakai, T. *et al.*: Toshiba BRIDJE at NTCIR-4 CLIR: Monolingual/Bilingual IR and Flexible Feedback, *NTCIR-4 CLIR Working Notes*, to appear (2004).

[9] Sakai, T. *et al.*: Toshiba ASKMi at NTCIR-4 QAC2 *NTCIR-4 QAC2 Working Notes*, to appear (2004).

[10] Voorhees, E. M.: Building A Question Answering Test Collection, *ACM SIGIR 2000 Proceedings*, pp. 200–207, 2000.

[11] Voorhees, E. M.: Overview of the TREC 2001 Question Answering Track, *TREC 2001 Proceedings*, 2001.

[12] Voorhees, E. M.: Overview of the TREC 2002 Question Answering Track, *TREC 2002 Proceedings*, 2002.