PERFORMANCE MEASURES FOR INFORMATION EXTRACTION

John Makhoul, Francis Kubala, Richard Schwartz, Ralph Weischedel

BBN Technologies, GTE Corp.
Cambridge, MA 02138
{makhoul, fkubala, schwartz, weischedel}@bbn.com

ABSTRACT

While precision and recall have served the information extraction community well as two separate measures of system performance, we show that the *F*-measure, the weighted harmonic mean of precision and recall, exhibits certain undesirable behaviors. To overcome these limitations, we define an error measure, the *slot error rate*, which combines the different types of error directly, without having to resort to precision and recall as preliminary measures. The slot error rate is analogous to the word error rate that is used for measuring speech recognition performance; it is intended to be a measure of the cost to the user for the system to make the different types of errors.

1. INTRODUCTION

Precision (*P*) and *recall* (*R*) have been used regularly to measure the performance of information retrieval and information extraction systems. Precision deals with substitution and insertion errors while recall deals with substitution and deletion errors. Because of the community's desire to have a single measure of performance that deals with all three types of errors simultaneously – substitutions, deletions, and insertions – a single figure of merit, the *F-measure*, has been defined as a weighted combination of *P* and *R*. In this paper, we analyze the *F*-measure in detail and show some of its properties and limitations. The discussion leads naturally to a new error measure that overcomes these limitations. The proposed error measure is also consistent with the error measure already in use by the speech recognition community.

Even though the conclusions are applicable to various information extraction and retrieval problems, we shall use primarily examples from the information extraction tasks as defined in the Message Understanding Conference (MUC) evaluations [1].

2. PERFORMANCE MEASURES

For a given test, we assume that there is a *reference* comprising a set of tags representing ground truth. Each tag consists of one or more slots, depending on the tag. (For example, in Named-Entity extraction in MUC, each tag has two slots "type" and "extent" while in Template-Element and Scenario-Template extraction, the tags can have anywhere from one to six slots [1].) Each system participating in the test produces a response or *hypothesis* comprising another set of tags, each of which also consists of one or more slots. An algorithm is then used to align the hypothesis against the reference. The corresponding slots are then matched and scored as either correct or not. If not correct, the error is marked as either a substitution (incorrect slot), deletion (missing slot), or insertion (spurious slot). The scores are then added up and different measures of performance are computed.

We should point out that this paper does not address the important issues of how to align the hypothesis to the reference or how to decide whether a slot is correct or not. This paper is only concerned with how to compute system performance, once the alignment is completed and the correct/incorrect decisions for all the slots have been made.

To help in the analysis that follows, we define the following symbols:

N = total number of slots in the reference

M = total number of slots in the hypothesis

C = number of correct slots – those slots in the hypothesis that align with slots in the reference and are scored as correct

S = number of substitutions (incorrect slots) – slots in the hypothesis that align with slots in the reference and are scored as incorrect

D = number of deletions (missing slots or false rejections) – slots in the reference that do not align with any slots in the hypothesis

I = number of insertions (spurious slots or false acceptances) – slots in the hypothesis that do not align with any slots in the reference.

It is clear from the above definitions that

$$N = C + S + D \tag{1}$$

$$M = C + S + I. \tag{2}$$

The total number of correct, substitution, and deleted slots is equal to the total number of slots in the reference, *N*, which is fixed for a given test set. The value of *M*, however, is in general different for each system being tested. *M* may be larger or smaller than *N*, depending on whether insertions are more or less than deletions.

Precision and recall are then defined by:

$$P = \frac{C}{M} = \frac{C}{C + S + I}, \quad (3)$$

$$R = \frac{C}{N} = \frac{C}{C + S + D}. \quad (4)$$

Precision is the percentage of slots in the hypothesis that are correct, while recall is the percentage of reference slots for which the hypothesis is correct. Precision takes account of substitution and insertion errors while recall takes account of substitution and deletion errors.

In the interest of having a single performance measure, the *F*-measure is used; it is defined as the weighted harmonic mean of *P* and *R* [2]:[1]

$$F = \left[\frac{\alpha}{P} + \frac{1-\alpha}{R}\right]^{-1} = \frac{PR}{(1-\alpha)P + \alpha R}, 0 \le \alpha \le 1. \quad (5a)$$

The most popular value corresponds to α = 0.5 and *F* reduces to [1]:

$$F = \frac{2PR}{P + R}, \alpha = 0.5. \quad (5b)$$

Substituting (3) and (4) in (5a), we obtain

$$F = \frac{C}{C + S + (1-\alpha)D + \alpha I} = \frac{C}{(1-\alpha)N + \alpha M}, 0 \le \alpha \le 1. \quad (6)$$

Since *F* is a figure of merit, the higher its value the better we consider the performance of the system. We can then define *E* = 1 – *F* as a corresponding "error measure". From (6) we see that

$$E = 1 - F = \frac{S + (1-\alpha)D + \alpha I}{C + S + (1-\alpha)D + \alpha I} \quad (7)$$

$$= \frac{S + (1-\alpha)D + \alpha I}{(1-\alpha)N + \alpha M}, 0 \le \alpha \le 1.$$

Note that *P*, *R*, *F*, and *E* are all guaranteed to be between 0 and 1.

For α = 0.5, *E* in (7) reduces to:

$$E = \frac{S + (D + I)/2}{C + S + (D + I)/2} = \frac{S + (D + I)/2}{(N + M)/2}. \quad (8)$$

[1] In his original proposal, van Rijsbergen [2, p. 174] defines the combination function *F* as 1 minus the term in (5a), which he renames immediately as the effectiveness measure *E*, which corresponds to the error measure *E* in (7). This effectiveness measure appears to be the main measure used in the information retrieval literature. We use the term *F* here as it has been used in the MUC evaluations [1] but we substitute $\alpha = 1/(1+\beta^2)$ which is in van Rijsbergen's original definition.

3. ANALYSIS

The denominator in (8) is equal to the average of the number of slots in the reference and in the hypothesis. But the major effect in (8) is the fact that, in the numerator, the deletion and insertion errors are cut (or deweighted) by a factor of two! If our objective is to count all errors, then there is no *a priori* reason why we should deweight deletions and insertions in this manner. In other words, by simply using *F* as our performance measure, we are implicitly discounting our overall error rate, making our systems look like they are much better than they really are!

It is important to note that the definitions of *P* and *R* are quite adequate as separate measures of system performance. It is when *P* and *R* are fused into a single measure, as in (5), that the problem arises. For $0 < \alpha < 1$, this fusion between *P* and *R* causes both deletions and insertions to be deweighted in *E*. Indeed, the numerator in (7) contains the convex sum of *D* and *I* which can never be greater than either. In other words, no matter what weight α is chosen to combine *P* and *R*, the deweighting of *D* and *I* relative to *S* is guaranteed. Below, we examine other single performance measures that focus directly on the three types of error instead of relying on *P* and *R* as primary measures.

4. MUC ERROR MEASURE

A possible solution to the problem described above is provided by the error measure ERR defined by MUC [1]:

$$\text{ERR} = \frac{S + D + I}{C + S + D + I}. \quad (9)$$

ERR removes the deweighting of *D* and *I* by simply removing the α weights in (7). Indeed, ERR provides a big step in the right direction. We, therefore, find it curious that, even though ERR is computed in MUC evaluations, it has not been used as a primary measure of system performance. The reason may have been simply the historical inertia of first using *P* and *R*, and then *F* as measures of goodness, rather than using error metrics to measure system performance.

The reason for using error metrics to measure system performance is that error metrics represent the cost to the user in having the system make those errors. Cutting the error rate by a factor of two, for example, is an indication that the cost to the user is also cut in half in that, if the user were to correct those errors, one would have to devote only half as much effort. Improvements in system performance can then be tracked by measuring the relative decrease in error rate.

The definition of ERR, however, still has a problem in that it implicitly deweights insertion errors relative to deletions and substitutions. This fact becomes more obvious when we rewrite (9) as

$$\text{ERR} = \frac{S + D + I}{N + I}. \quad (10)$$

For a given test set, *N* is fixed. It is clear from (10) that ERR is a

linear function of S and D but it is a nonlinear function of I. The nonlinearity is compressive in I in that an increase in I increases ERR by a smaller amount than a similar increase in either S or D. The reason, of course, is that the denominator increases when we increase I but does not increase when either S or D are increased. (Even though N is a function of S and D, an increase in either of them must be balanced by a decrease in the other two parameters so that the sum $C + S + D = N$ remains constant, equal to the total number of slots in the reference.) One can also show that P has the same compressive property in I.

5. PROPOSED ERROR MEASURE

Our proposed solution to the deweighting of insertions problem in ERR is to simply remove I from the denominator in (10). The result is what we shall call the *slot error rate*, SER, defined as:

$$\text{SER} = \frac{S + D + I}{N} = \frac{S + D + I}{C + S + D}$$

$$= \frac{\text{Total number of slot errors}}{\text{Total number of slots in reference}}. \qquad (11)$$

SER is simply the ratio of the total number of slot errors – substitutions, deletions, and insertions – divided by the total number of slots in the reference, which is fixed for a given test. In this way, the errors from all systems are compared against a fixed base. Since N is fixed, SER is a linear function of S, D, and I.

For particular applications, certain types of error may be deemed more or less important than others. In that case, the definition of the error SER in (11) can be modified by multiplying the different types of errors by different weights. However, for simply developing the information extraction technology, we see no compelling reason to weight one type of error more or less than the others.

The slot error rate SER is exactly analogous to the word error rate which has been in use as the primary measure of speech recognition performance for many years. The word error rate is similarly defined as the sum of word substitutions, deletions, and insertions, divided by the total number of words in the reference. The simplicity and utility of this error in measuring the relative improvements in speech recognition performance has withstood the test of many years of significant advances in the state of the art.

6. COMPARATIVE ANALYSIS OF ERROR MEASURES

ERR, the error measure defined by MUC, does have one esthetic advantage in that it is guaranteed to be between 0 and 1, while SER, the slot error rate, can become greater than 1 under certain high error conditions. Some may feel uncomfortable with the notion of an error rate that is greater than 100%, but this possibility is not as unreasonable as it might appear at first glance.

Consider the following two hypothetical high-error-rate systems: System A gives nothing as its output, while system B produces $0.2N$ slots that are all judged as insertions. So, system A will have $S = 0$, $D = N$, and $I = 0$, and system B will have $S = 0$, $D = N$, and $I = 0.2N$. Table 1 shows the two cases and the corresponding values of ERR and SER. Both error measures give system A an error rate of 100%,which is appropriate since the system missed all N slots in the reference. But system B not only missed all N slots in the reference, it also introduced an additional $0.2N$ insertions. ERR gives system B an error of 100% also, effectively saying that system A and B performed equally, while the error SER says that system B is 20% worse than system A and, since it already gave system A 100%, it should give system B a score of 120%. We believe that SER gives the more satisfying answer in terms of the cost to the user. Not only did system B miss all the useful information, it introduced an additional amount of incorrect information that the user has to deal with.

	S	*D*	*I*	ERR	SER
System A	0	N	0	1.0	1.0
System B	0	N	0.2N	1.0	1.2

Table 1. The values of two error measures for two high-error-rate systems.

The above example is quite extreme, of course. To see the effects of the different error measures on more realistic data, we show in Table 2 the values of the error measures for three different types of information extraction tests from the MUC-6 evaluations: Named Entity, Template Element, and Scenario Template [1]. For each test, we show the results for the best performing system in MUC-6. The values shown are those of the F-measure in (5b), E, ERR, and SER, all multiplied by 100. In all cases

$$\text{SER} \geq \text{ERR} \geq E$$

as is guaranteed from the definitions of the three error measures. The major difference, however, takes place when we go from E to ERR; the increase from ERR to SER is not as large. The reason for the large increase in going from E to ERR is the fact that ERR does not deweight the deletions and insertions by a factor of 2. While Table 1 showed an example where SER could be much larger than ERR, we see from Table 2 that, in practice, the difference is not so large. By comparing the E and SER columns, we see that the values of E are approximately 30% lower than those of SER. Our interpretation is that E, and therefore F, under represents the total error by about 30%. Based on these results, we can write

$$1 - F \cong 0.7\ \text{SER}$$
$$\text{SER} \cong 1.5(1 - F). \qquad (12)$$

In other words, in practice, the slot error rate is about 50% higher than the error rate represented by the F-measure.

MUC-6 Test	*F*	*E* = 1 - *F*	ERR	SER
Named Entity	96.42	3.58	5.01	5.07
Template Element	79.99	20.01	29.46	30.80
Scenario Template	56.40	43.60	56.52	61.17

Table 2. The values of the *F*-measure along with three different error measures: *E*, ERR, and SER, for the best performing system in each of three different information extraction tests from MUC-6.

7. CONCLUSIONS

Precision and recall have been and continue to be very useful measures of performance for information retrieval and extraction. Precision deals with substitution and insertion errors while recall deals with substitution and deletion errors. Because of our desire to have a single measure of performance that deals with all three types of errors simultaneously, the *F*-measure was defined. By examining $E = 1 - F$, we showed how deletions and insertions are deweighted such that the combined error is never greater than either. To ameliorate this drawback of the *F*-measure, we proposed a simple error measure that is equal to the sum of the three types of errors – substitutions, deletions, and insertions – divided by the total number of slots in the reference. This error measure, the *slot error rate* SER, is analogous to the word error rate that is used worldwide for measuring speech recognition performance. As the community embarks on performing information extraction from speech, it is good to know that the same error measure is appropriate for assessing the performance of both technologies – speech recognition and information extraction.

REFERENCES

1. N. Chinchor and G. Dungca, "Four Scores and Seven Years Ago: The Scoring Method for MUC-6," *Proc. MUC-6 Conference*, Columbia, MD, pp. 33-38 and pp. 293-316, Nov. 1995.
2. C.J. van Rijsbergen, *Information Retrieval*, London: Butterworth, 1979.