MUC-4 EVALUATION METRICS

Nancy Chinchor, Ph.D.
Science Applications International Corporation
10260 Campus Point Drive, M/S A2-F
San Diego, CA 92121
chinchor@esosun.css.gov
(619) 458-2614

INTRODUCTION

The MUC-4 evaluation metrics measure the performance of the message understanding systems. This paper describes the scoring algorithms used to arrive at the metrics as well as the improvements that were made to the MUC-3 methods. MUC-4 evaluation metrics were stricter than those used in MUC-3. Given the differences in scoring between MUC-3 and MUC-4, the MUC-4 systems' scores represent a larger improvement over MUC-3 performance than the numbers themselves suggest.

The major improvements in the scoring of MUC-4 were the automation of the scoring of set fill slots, partial automation of the scoring of string fill slots, content-based mapping enforced across the board, the focus on the ALL TEMPLATES score as opposed to the MATCHED/MISSING score in MUC-3, the exclusion of the template id scores from the score tallies, and the addition of the object level scores, string fills only scores, text filtering scores, and F-measures. These improvements and their effects on the scores are discussed in detail in this paper.

SCORE REPORT

The MUC-4 Scoring System produces score reports in various formats. These reports show the scores for the templates and messages in the test set. Varying amounts of detail can be reported. The scores that are of the most interest are those that appear in the comprehensive summary report. Figure 1 shows a sample summary score report. The rows and columns of this report are explained below.

Scoring Categories

The basic scoring categories are located at the top of the score report. These categories are defined in Table 1. The scoring program determines the scoring category for each system response. Depending on the type of slot being scored, the program can either determine the category automatically or prompt the user to determine the amount of credit the response should be assigned.

- If the response and the key are deemed to be equivalent, then the fill is assigned the category of correct (COR).
- If partial credit can be given, the category is partial (PAR).
- If the key and response simply do not match, the response is assigned an incorrect (INC).
- If the key has a fill and the response has no corresponding fill, the response is missing (MIS).
- If the response has a fill which has no corresponding fill in the key, the response is spurious (SPU).
- If the key and response are both left intentionally blank, then the response is scored as noncommittal (NON).

SLOT	POS	ACT	COR	PAR	INC	ICR	IPA	SPU	MIS	NON	REC	PRE	OVG	FAL
template-id	10	10	8	0	0	0	0	2	2	4	80	80	20	
inc-date	10	8	7	0	1	0	0	0	2	0	70	88	0	
inc-loc	9	8	4	3	0	0	0	1	2	0	61	69	12	
inc-type	10	8	8	0	0	0	0	0	2	0	80	100	0	0
inc-stage	10	8	7	0	1	0	0	0	2	0	70	88	0	5
inc-instr-id	5	5	4	0	1	0	0	0	0	5	80	80	0	
inc-instr-type	6	6	4	1	1	0	0	0	0	4	75	75	0	1
perp-inc-cat	6	5	5	0	0	0	0	0	1	4	83	100	0	0
perp-ind-id	5	3	3	0	0	0	0	0	2	5	60	100	0	
perp-org-id	2	2	2	0	0	0	0	0	0	8	100	100	0	
perp-org-conf	3	3	3	0	0	0	0	0	0	7	100	100	0	0
phys-tgt-id	8	6	5	0	1	0	0	0	2	2	62	83	0	
phys-tgt-type	8	6	5	0	1	0	0	0	2	2	62	83	0	1
phys-tgt-num	8	6	5	0	1	0	0	0	2	2	62	83	0	
phys-tgt-nation	2	1	1	0	0	0	0	0	1	8	50	100	0	0
phys-tgt-effect	6	5	3	0	2	0	0	0	1	4	50	60	0	4
phys-tgt-total-num	0	0	0	0	0	0	0	0	0	10	*	*	*	
hum-tgt-name	1	1	1	0	0	0	0	0	0	9	100	100	0	
hum-tgt-desc	6	6	5	0	1	0	0	0	0	5	83	83	0	
hum-tgt-type	6	6	4	0	2	0	0	0	0	5	67	67	0	2
hum-tgt-num	6	6	5	0	1	0	0	0	0	5	83	83	0	
hum-tgt-nation	2	2	1	0	1	0	0	0	0	9	50	50	0	0
hum-tgt-effect	4	4	2	1	1	0	0	0	0	7	62	62	0	1
hum-tgt-total-num	0	0	0	0	0	0	0	0	0	10	*	*	*	
inc-total	50	43	34	4	4	0	0	1	8	9	72	84	2	
perp-total	16	13	13	0	0	0	0	0	3	24	81	100	0	
phys-tgt-total	32	24	19	0	5	0	0	0	8	28	59	79	0	
hum-tgt-total	25	25	18	1	6	0	0	0	0	50	74	74	0	
MATCHED/MISSING	123	105	84	5	15	0	0	1	19	111	70	82	1	
MATCHED/SPURIOUS	106	125	84	5	15	0	0	21	2	108	82	69	17	
MATCHED ONLY	106	105	84	5	15	0	0	1	2	82	82	82	1	
ALL TEMPLATES	123	125	84	5	15	0	0	21	19	137	70	69	17	
SET FILLS ONLY	63	54	43	2	9	0	0	0	9	50	70	81	0	0
STRING FILLS ONLY	27	23	20	0	3	0	0	0	4	34	74	87	0	
TEXT FILTERING	7	8	7	*	*	*	*	1	0	4	100	88	12	20
F-MEASURES									P&R		2P&R		P&2R	
									69.5		69.2		69.8	

Figure 1: Sample Score Report

❑ **Correct**	**response = key**
❑ **Partial**	**response = key**
❑ **Incorrect**	**response = key**
❑ **Non-committal**	**key and response are both blank**
❑ **Spurious**	**key is blank and response is not**
❑ **Missing**	**response is blank and key is not**

Table 1: Scoring Categories

In Figure 1, the two columns titled ICR (interactive correct) and IPA (interactive partial) indicate the results of interactive scoring. Interactive scoring occurs when the scoring system finds a mismatch that it cannot automatically resolve. It queries the user for the amount of credit to be assigned. The number of fills that the user assigns to the category of correct appears in the ICR column; the number of fills assigned partial credit by the user appears in the IPA column.

In Figure 1, the two columns labelled possible (POS) and actual (ACT) contain the tallies of the numbers of slots filled in the key and response, respectively. Possible and actual are used in the computation of the evaluation metrics. Possible is the sum of the correct, partial, incorrect, and missing. Actual is the sum of the correct, partial, incorrect, and spurious.

Evaluation Metrics

The evaluation metrics were adapted from the field of Information Retrieval (IR) and extended for MUC. They measure four different aspects of performance and an overall combined view of performance. The four evaluation metrics of recall, precision, overgeneration, and fallout are calculated for the slots and summary score rows (see Table 2). These are listed in the four rightmost columns of the score report in Figure 1. The fifth metric, the F-measure, is a combined score for the entire system and is listed at the bottom of the score report.

recall	=	$\frac{\text{correct} + (\text{partial} \times 0.5)}{\text{possible}}$
precision	=	$\frac{\text{correct} + (\text{partial} \times 0.5)}{\text{actual}}$
over-generation	=	$\frac{\text{spurious}}{\text{actual}}$
fallout	=	$\frac{\text{incorrect} + \text{spurious}}{\text{possible incorrect}}$

Table 2: Evaluation Metrics

Recall (REC) is the percentage of possible answers which were correct. Precision (PRE) is the percentage of actual answers given which were correct. A system has a high recall score if it does well relative to the number of slot fills in the key. A system has a high precision score if it does well relative to the number of slot fills it attempted.

In IR, a common way of representing the characteristic performance of systems is in a precision-recall graph. Normally as recall goes up, precision tends to go down and vice versa [1]. One approach to improving recall is to increase the system's generation of slot fills. To avoid overpopulation of the template database by the message understanding systems, we introduced the measure of overgeneration. Overgeneration (OVG) measures the percentage of the actual attempted fills that were spurious.

Fallout (FAL) is a measure of the false positive rate for slots with fills that come from a finite set. Fallout is the tendency for a system to choose incorrect responses as the number of possible responses increases. Fallout is calculated for all of the set fill slots listed in the score report in Figure 1 and is shown in the last column on the right. Fallout can be calculated for the SET FILLS ONLY row because that row contains the summary tallies for the set fill slots. The TEXT FILTERING row discussed later contains a score for fallout because the text filtering problem also has a finite set of responses possible.

These four measures of recall, precision, overgeneration, and fallout characterize different aspects of system performance. The measures of recall and precision have been the central focus for analysis of the results. Overgeneration is a measure which should be kept under a certain value. Fallout was rarely used in the analyses done of the results. It is difficult to rank the systems since the measures of recall and precision are often equally important yet negatively correlated. In IR, a method was developed for combining the measures of recall and precision to get a single measure. In MUC-4, we use van Rijsbergen's F-measure [1, 2] for this purpose.

The F-measure provides a way of combining recall and precision to get a single measure which falls between recall and precision. Recall and precision can have relative weights in the calculation of the F-measure giving it the flexibility to be used for different applications. The formula for calculating the F-measure is

$$F = \frac{(\beta^2 + 1.0) \times P \times R}{\beta^2 \times P + R}$$

where P is precision, R is recall, and β is the relative importance given to recall over precision. If recall and precision are of equal weight, $\beta = 1.0$. For recall half as important as precision, $\beta = 0.5$. For recall twice as important as precision, $\beta = 2.0$.

The F-measure is higher if the values of recall and precision are more towards the center of the precision-recall graph than at the extremes and their sums are the same. So, for $\beta = 1.0$, a system which has recall of 50% and precision of 50% has a higher F-measure than a system which has recall of 20% and precision of 80%. This behavior is exactly what we want from a single measure.

The F-measures are reported in the bottom row of the summary score report in Figure 1. The F-measure with recall and precision weighted equally is listed as "P&R." The F-measure with precision twice as important as recall is listed as "2P&R." The F-measure with precision half as important as recall is listed as "P&2R." The F-measure is calculated from the recall and precision values in the ALL TEMPLATES row. Note that the recall and precision values in the ALL TEMPLATES row are rounded integers and that this causes a slight inaccuracy in the F-measures. The values used for calculating statistical significance of results are floating point values all the way through the calculations. Those more accurate values appear in the paper "The Statistical Significance of the MUC-4 Results" and in Appendix G of these proceedings.

Summary Rows

The four rows labeled "inc-total," "perp-total," "phys-tgt-total," and "hum-tgt-total" in the summary score report in Figure 1 show the subtotals for associated groups of slots referred to as "objects." These are object level scores for the incident, perpetrator, physical target, and human target. They are the sums of the scores shown for the

individual slots associated with the object as designated by the first part of the individual slot labels. The template for MUC-4 was designed as a transition from a flat template to an object-oriented template. Although referred to as object-oriented, the template is not strictly object-oriented, but rather serves as a data representation upon which an object-oriented system could be built[3]. However, no object-oriented database system was developed using this template as a basis.

The four summary rows in the score report labelled "MATCHED/MISSING," "MATCHED/SPURIOUS," "MATCHED ONLY," and "ALL TEMPLATES" show the accumulated tallies obtained by scoring spurious and missing templates in different manners. Each message can cause multiple templates to be generated depending on the number of terrorist incidents it reports. The keys and responses may not agree in the number of templates generated or the content-based mapping restrictions may not allow generated key and response templates to be mapped to each other. These cases lead to spurious and/or missing templates. There are differing views as to how much systems should be penalized for spurious or missing templates depending upon the requirements of the application. These differing views have lead us to provide the four ways of scoring spurious and missing information as outlined in Table 3.

- **Matched Only**
  - *Missing and spurious templates scored in template-id slot only*
- **Matched/Missing**
  - *Missing template slots scored as missing*
  - *Spurious templates scored only in template-id slot*
- **Matched/Spurious**
  - *Spurious template slots scored as spurious*
  - *Missing templates scored only in template-id slot*
- **All Templates**
  - *Missing template slots scored as missing*
  - *Spurious template slots scored as spurious*

Table 3: Manners of Scoring

The MATCHED ONLY manner of scoring penalizes the least for missing and spurious templates by scoring them only in the template id slot. This template id score does not impact the overall score because the template id slot is not included in the summary tallies; the tallies only include the other individual slots. The MATCHED/MISSING method scores the individual slot fills that should have been in the missing template as missing and scores the template as missing in the template id slot; it does not penalize for slot fills in spurious templates except to score the spurious template in the template id slot. MATCHED/SPURIOUS, on the other hand, penalizes for the individual slot fills in the spurious templates, but does not penalize for the missing slot fills in the missing templates. ALL TEMPLATES is the strictest manner of scoring because it penalizes for both the slot fills missing in the missing templates and the slots filled in the spurious templates. The metrics calculated based on the scores in the ALL TEMPLATES row are the official MUC-4 scores.

These four manners of scoring provide four points defining a rectangle on a precision-recall graph which we refer to as the "region of performance" for a system (see Figure 2). At one time, we thought that it would be useful to

compare the position of the center of this rectangle across systems, but later realized that two systems could have the same centers but very different size rectangles. Plotting the entire region of performance for each system does provide a useful comparison of systems.

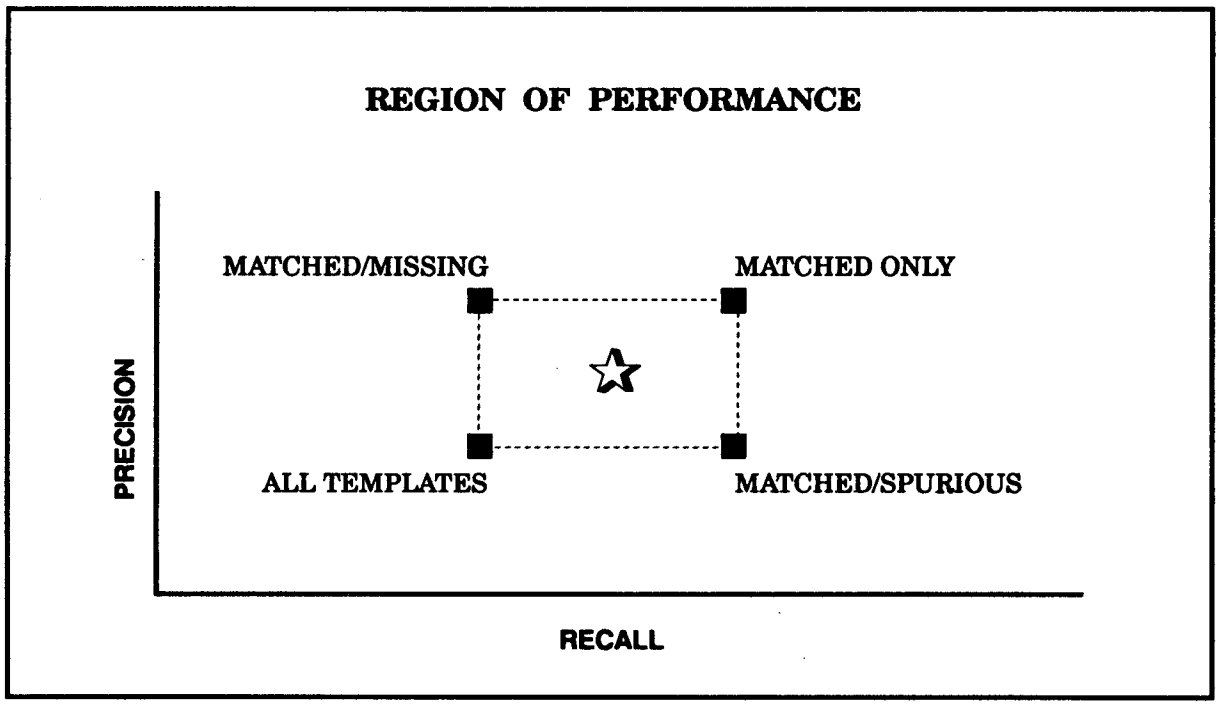

Figure 2: Region of Performance

In Figure 1, the score report contains two summary rows (SET FILLS ONLY and STRING FILLS ONLY) which give tallies for a subset of the slots based on the type of fill the slot can take. These rows give tallies that show the system's performance on these two types of slots: set fill slots and string fill lots. Set fill slots take a fill from a finite set specified in a configuration file. String fill slots take a fill that is a string from a potentially infinite set.

Text Filtering

The purpose of the text filtering row is to report how well systems distinguish relevant and irrelevant messages. The scoring program keeps track of how many times each of the situations in the contingency table arises for a system (see Table 4). It then uses those values to calculate the entries in the TEXT FILTERING row. The evaluation metrics are calculated for the row as indicated by the formulas at the bottom of Table 4. An analysis of the text filtering results appears elsewhere in these proceedings.

IMPROVEMENTS OVER MUC-3

The major improvements in the scoring of MUC-4 included:

- automating the scoring as effectively as possible
- restricting the mapping of templates to cases where particular slots matched in content as opposed to mapping only according to an optimized score
- a focus on the ALL TEMPLATES score as opposed to the MATCHED/MISSING score in MUC-3

- the exclusion of template id scores from the summary score tallies
- the inclusion of more summary information including object level scores, string fills only scores, text filtering scores, and F-measures.

These changes are interdependent; they interact in ways that affect the overall scores of systems and serve to make MUC-4 a more demanding evaluation than MUC-3.

	Relevant is Correct	Irrelevant is Correct	
Decides Relevant	a	b	a+b
Decides Irrelevant	c	d	c+d
	a+c	b+d	a+b+c+d = n

	POS	ACT	COR	PAR	INC	ICR	IPA	SPU	MIS	NON
Text Filtering	a+c	a+b	a	-	-	-	-	b	c	d

Recall = a/(a+c) **Overgeneration = b/(a+b)**

Precision = a/(a+b) **Fallout = b/(b+d)**

Table 4: Text Filtering

The complete automation of the scoring of set fill slots was possible due to the information in a slot configuration file which told the program the hierarchical structure of the set fills. If a response exactly matches the key, it is scored as correct. If a response is a more general set fill element than the key according to the pre-specified hierarchy, it is scored as partially correct. If the response cannot be scored as correct or partially correct by these criteria then the set fill is scored as incorrect. All set fills can thus be automatically scored. Often, however, the set fill is cross-referenced to another slot which is a string fill. The scoring of string fills cannot be totally automated. Instead the scoring program refers to the history of the interactive scoring of the cross-referenced slot, and with that information, it then determines the score for the set fill slot which cross-references the string fill slot.

The scoring of the string fill slots was partially automated by using two methods. In the first method, used for mapping purposes, strings were considered correct if there was a one-word overlap and the word was not from a short list of premodifiers. In the second method, used for scoring purposes, some mismatching string fills could be matched automatically by stripping these premodifiers from the key and response and seeing if the remaining material matched. Other mismatching string fills caused the user to be queried for the score. The automation of the set fill and string fill scoring was critical to the functioning of the content-based mapping.

The content-based mapping restrictions were added to MUC-4 to prevent fortuitous mappings which occurred in MUC-3. In MUC-3, templates were mapped to each other based on a simple optimization of scores. Sometimes the optimal score was the result of a lucky mapping which was not really the most appropriate mapping.

Certain slots such as incident type were considered essential for the mapping to occur in MUC-4. The mapping restrictions can be specified in the scorer's configuration file using a primitive logic. For the MUC-4 testing, the templates must have at least a partial match on the incident type and at least one of the following slots:

- physical target identifier
- physical target type
- human target name
- human target description
- human target type
- perpetrator individual identifier
- perpetrator organization identifier

The content-based mapping restrictions could result in systems with sparse templates having few or no templates mapped. When a template does not map, the result is one missing and one spurious template. This kind of penalty is severe when the ALL TEMPLATES row is the official score, because the slots in the unmapped templates all count against the system as either missing or spurious. This aspect of the scoring was one of the main reasons that MUC-4 was more demanding than MUC-3.

The focus on the ALL TEMPLATES score as opposed to the MATCHED/MISSING score in MUC-3 meant that the strictest scores for a system were its official scores. So even if a system's official scores were the same for MUC-3 and MUC-4, the system had improved in MUC-4. Additionally, the scores for the template id row were not included in the summary row tallies in MUC-4 as they had been in MUC-3. Previously, systems were getting extra credit for the optimal mapping. This bonus was taken away by the exclusion of the template id scores from the score tallies in MUC-4.

In addition to the more demanding scoring, MUC-4 also measured more information about system performance. Object level scores were added to see how well the systems did on different groupings of slots concerning the incident, perpetrator, physical target, and human target. Also, the score for the string fill slots was tallied as a comparison with the score for set fill slots that was already there in MUC-3. The text filtering scores gave additional information on the capabilities of systems to determine relevancy. The F-measures combined recall and precision to give a single measure of performance for the systems.

SUMMARY

The evaluation metrics used in MUC-4 gave a stricter and more complete view of the performance of the systems than the metrics used in MUC-3. The improved overall numerical scores of the systems under these more difficult scoring conditions indicate that the state of the art has moved forward with MUC-4.

REFERENCES

[1] Frakes, W.B.and R. Baeza-Yates (eds.) (1992) Information Retrieval: Data Structures & Algorithms. Englewood Cliffs: Prentice Hall.

[2] Van Rijsbergen, C.J. (1979) Information Retrieval. London: Butterworths.

[3] Nierstrasz, Oscar (1989) "A Survey of Object-Oriented Concepts" in W. Kim and F. H. Lochovsky (Eds.) Object-Oriented Concepts, Databases, and Applications. New York: Addison-Wesley.