MUC-5 EVALUATION METRICS

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

Beth Sundheim
Naval Command, Control, and Ocean Surveillance Center
RDT&E Division (NRaD)
Information Access Technology Project Team, Code 44208
San Diego, CA 92152-7420
sundheim@nosc.mil

INTRODUCTION

The metrics used for the Fifth Message Understanding Conference (MUC-5) evaluation are a major update to those used for MUC-4 in 1992. The official MUC-5 metrics express error rates while the official MUC-4 metrics express performance in terms of recall and precision (used for MUC-5 only as "unofficial" metrics). This paper discusses the current metrics and the reasons for their adoption.

SCORE REPORTS

The MUC-5 Scoring System is evaluation software that aligns and scores the templates produced by the information extraction systems under evaluation in comparison to an "answer key" created by humans. The Scoring System produces comprehensive summary reports showing the overall scores for the templates in the test set; these may be supplemented by detailed score reports showing scores for each template individually. Figure 1 shows a sample summary score report in the joint ventures task domain for the error metrics; Figure 2 shows a corresponding summary score report for the recall-precision metrics.

Scoring Categories

The basic scoring categories are found in the score report under the column headings COR, PAR, INC, XCR, XPA, XIC, MIS, SPU, and NON. These categories have not fundamentally changed since the MUC-4 evaluation. The rows in the body of the score report are for the various slots and objects in the template; various totals appear at the bottom.

For the MUC-5 evaluation, alignment of system responses (i.e., templates, objects, and slot-fillers generated by the system under evaluation) with the answer key was done fully automatically, and scoring was done interactively. In interactive scoring mode, the evaluator is queried for a scoring decision only under certain circumstances; under most circumstances, the scoring decisions are made automatically. The meaning of each of the scoring categories is described below and summarized in Table 1.

- If the response and the key are deemed to be equivalent, the category is correct (COR); if interactively assigned, a tally appears in both the COR and XCR (interactive correct) columns.
- If the response and the key are judged to be a near match, the category is partial (PAR); if interactively assigned, a tally appears in both the PAR and XPA (interactive partial) columns.

SLOT	POS	ACT	COR	PAR	INC	XCR	XPA	XIC	SPU	MIS	NON	ERR	UND	OVG	SUB
<template>	282	282	282	0	0	0	0	0	0	0	0	0	0	0	0
content	348	389	289	0	6	0	0	0	94	53	11	35	15	24	2
subtotals	348	389	289	0	6	0	0	0	94	53	11	35	15	24	2
<tie-up-relati	348	389	289	0	6	0	0	0	94	53	11	35	15	24	2
status	348	389	231	0	64	0	0	0	94	53	0	48	15	24	22
entity	791	834	535	0	94	0	0	0	305	162	0	51	20	33	15
joint-venture	180	212	101	0	23	0	0	0	88	56	96	62	31	42	18
ownership	103	122	64	0	4	0	0	0	54	35	180	59	34	44	6
activity	387	367	214	0	44	0	0	0	109	129	10	57	33	30	17
subtotals	1809	2024	1145	0	229	0	0	0	650	435	286	53	24	32	17
<entity>	976	1057	749	0	61	0	0	0	247	166	0	39	17	23	8
name	872	937	554	44	115	4	38	81	224	159	27	47	18	24	19
aliases	359	389	232	6	5	0	5	3	146	116	399	53	32	38	3
location	322	338	140	19	23	0	3	1	156	140	443	69	43	46	18
nationality	265	212	100	0	11	0	0	9	101	154	509	73	58	48	10
type	976	1057	716	0	94	0	0	0	247	166	0	41	17	23	12
⋮															
ALL OBJECTS	12125	13913	6793	149	1562	6	85	124	5405	3621	3996	61	30	39	19
MATCHED ONLY	9140	9729	6793	149	1159	6	85	124	1624	1039	3996	36	11	17	15

Richness-Normalized Error	Wrong	Req-fills	All-fills	Min-err	Max-err
	10662.5	11813	12138	0.8784	0.9026

Error Rate Per Word	Wrong	Word-count	Error-rate
	10662.5	92862	0.1148

Figure 1: Sample Error Score Report.

- If the key and response do not match, the category is incorrect (INC); if interactively assigned, a tally appears in both the INC and XIC (interactive incorrect) columns.
- If the key has a fill and the response has no corresponding fill, the category is missing (MIS).
- If the response has a fill which has no corresponding fill in the key, the category is spurious (SPU).
- If the key and response are both left blank, then the category is noncommittal (NON).

The columns in Figures 1 and 2 labelled possible (POS) and actual (ACT) contain the tallies of the number of slot fillers that should be generated and the number of fillers that the system under evaluation actually generated, respectively. Possible is the sum of the correct, partial, incorrect, and missing. Actual is the sum of the correct, partial, incorrect, and spurious. These tallies are used in the computation of some of the evaluation metrics. The total possible is system-dependent and is therefore computed by summing the tallies assigned to the system responses rather than by simply summing the slot fillers to be found in the key template. In contrast, a system-independent metric will be explained in a later section.

SLOT	POS	ACT	COR	PAR	INC	XCR	XPA	XIC	SPU	MIS	NON	REC	PRE	UND	OVG
<template>	282	282	282	0	0	0	0	0	0	0	0	100	100	0	0
content	348	389	289	0	6	0	0	0	94	53	11	83	74	15	24
subtotals	348	389	28	0	6	0	0	0	94	53	11	83	74	15	24
<tie-up-relati	348	389	289	0	6	0	0	0	94	53	0	83	74	15	24
status	348	389	231	0	64	0	0	0	94	53	0	66	59	15	24
entity	791	934	535	0	4	0	0	0	305	162	0	68	57	20	33
joint-venture	180	212	101	0	23	0	0	0	88	56	96	56	48	31	42
ownership	10	122	64	0	4	0	0	0	54	35	180	62	52	34	44
activity	387	367	214	0	44	0	0	0	109	129	10	55	58	33	30
subtotals	1809	2024	1145	0	229	0	0	0	650	435	286	63	56	24	32
<entity>	976	1057	749	0	61	0	0	0	247	166	0	77	71	17	23
name	872	937	554	44	115	4	38	81	224	159	27	66	61	18	24
aliases	359	389	232	6	5	0	5	3	146	116	399	65	60	32	38
location	322	338	140	19	23	0	3	1	156	140	443	46	44	43	46
nationality	265	212	100	0	11	0	0	9	101	154	509	38	47	58	48
type	97	1057	716	0	94	0	0	0	247	166	0	73	68	17	23
⋮															
ALL OBJECTS	12125	13913	6793	149	1562	6	85	124	5405	3621	3996	57	49	30	39
MATCHED ONLY	9140	9729	6793	149	1159	6	85	124	1624	1039	3996	75	70	11	17
TEXT FILTERING	251	262	242	*	*	*	*	*	20	9	11	96	92	4	8
F-MEASURES										P&R		2P&R		P&2R	
										52.75		50.66		55.02	

Figure 2: Sample Recall-Precision Score Report.

Summary Rows

The two summary rows in the score report labelled "ALL OBJECTS" and "MATCHED ONLY" show the accumulated tallies obtained by scoring spurious and missing objects in different manners. Templates may contain

❑ **Correct**	**response = key**
❑ **Partial**	**response ≅ key**
❑ **Incorrect**	**response ≠ key**
❑ **Spurious**	**key is blank and response is not**
❑ **Missing**	**response is blank and key is not**
❑ **Noncommittal**	**key and response are both blank**

Table 1: Scoring Categories.

more than one instance of a kind of object, e.g., more than one <entity> object. The keys and responses may not agree in the number of objects generated. These cases lead to spurious and/or missing objects. Opinions as to how much systems should be penalized for spurious or missing objects differ depending upon the requirements of the application in mind. These differing views have lead us to provide the two ways of scoring spurious and missing information as outlined in Table 2.

The MATCHED ONLY manner of scoring penalizes the least for missing and spurious objects by scoring them only in the object ID slot. This object ID score does not impact the overall score because the object ID slot is not included in the summary tallies; the tallies include only the individual slots. ALL OBJECTS is a stricter manner of scoring because it penalizes for both the slot fills missing in the missing objects and the slots filled in the spurious object. The metrics calculated based on the scores in the ALL OBJECTS row of the error score report are the official MUC-5 scores.

- **Matched Only**
  - *Missing and spurious objects scored in object slot only*
- **All Objects**
  - *Missing object slots scored as missing*
  - *Spurious object slots scored as spurious*

Table 2: Manners of Scoring.

Evaluation Metrics

The rightmost four columns in both the error score report and the recall-precision score report contain the scores for the evaluation metrics. These are computed for each object and slot in the template, and overall scores are shown at the bottom.

The primary evaluation metrics for MUC-5 have been changed from those used in previous MUC evaluations. The reasoning behind this change will be described in a later section. First, the formulas used to calculate the evaluation metrics on the score reports will be given.

Error Metrics

The error per response fill (ERR) is the official measure of MUC-5 system performance. This measure is calculated as the number wrong divided by the total (possible plus spurious) as shown in Table 3. It is dependent on the system because tallies change according to the amount of spurious data generated and according to how the system filled slots that have optional or alternate fills in the key. (See the discussion below on richness-normalized error metric.)

Table 3 also shows the computation of three secondary metrics -- undergeneration, overgeneration, and substitution -- which isolate the three elements constituting overall error. Undergeneration and overgeneration were in use for MUC-4 as well, and this is why they appear in both the error score report and the recall-precision score report. Those metrics are computed the same way for both reports. The substitution metric is new for MUC-5 and is found only in the error score report. The metric is not isolated in the recall-precision view on information extraction; this is because it is a (negative) factor in both recall and precision; in the error-based view, on the other hand, it is isolated as a distinct type of error. The reader should note that the denominator in each of the secondary metrics is different because each metric offers a distinct perspective on the errors that a system can make.

Primary Metric	Error per response fill = $\frac{\text{wrong}}{\text{total}} = \frac{\text{INC} + \text{PAR}/2 + \text{MIS} + \text{SPU}}{\text{COR} + \text{PAR} + \text{INC} + \text{MIS} + \text{SPU}}$
Secondary Metrics	Undergeneration = $\frac{\text{MIS}}{\text{POS}} = \frac{\text{MIS}}{\text{COR} + \text{PAR} + \text{INC} + \text{MIS}}$
	Overgeneration = $\frac{\text{SPU}}{\text{ACT}} = \frac{\text{SPU}}{\text{COR} + \text{PAR} + \text{INC} + \text{SPU}}$
	Substitution = $\frac{\text{INC} + \text{PAR}/2}{\text{COR} + \text{PAR} + \text{INC}}$

Table 3: System-dependent Error Metrics.

The error per response fill has been chosen as the primary measure reported for a system for this evaluation because developers now need to focus on the sources of errors, explain them, and remedy them to push the state of the art. For example, if System A has the raw scores shown in Figure 3, its error per response fill is calculated as follows:

wrong = INC + PAR/2 + MIS + SPU = 25 + 5 + 0 + 10 = 40

total = COR + PAR + INC + MIS + SPU = 10 + 10 + 25 + 0 + 10 = 55

wrong/total = 40/55 = 73%

While the error per response fill metric and the undergeneration, overgeneration, and substitution metrics are designed to suit the system developers' need for performance diagnostics, a different measure that is as independent of the system and the text sample as possible may be more useful in some other circumstances. The richness-normalized error measure is designed to measure errors relative to the amount of information to be extracted from the texts. This metric is shown in one of the summary rows at the bottom of the error score report.

	COR	PAR	INC	SPU	MIS	NON	ERR
SYSTEM A	10	10	25	10	0	35	73

Figure 3: System A.

Richness-normalized error is calculated by dividing the number of errors per word by the number of key fills per word. This calculation reduces to the number of errors divided by the fill-count. If a program manager is considering use of a system on a distinct class of documents from the ones the system was tested on, this measure will predict the number of errors the system will make, given the richness of the new set of documents.

Due to the optional and alternate fills in the key, there will be a range of fill-counts from the minimum number of fills required to the maximum number of fills allowed. The difference between the two numbers represent "discretionary" fills, i.e., ones that represent the ambiguity inherent in the text.[1] The formaulas for calculating the minimum and maximum richness-normalized error appear in Table 4.

1. For further information on the variability inherent in the key templates, please refer to the published version of the proceedings, which will contain a paper about the text and template corpora.

Richness-Normalized Error	Minimum Error =	$\frac{\text{wrong}}{\text{All-fills}} = \frac{\text{INC} + \text{PAR}/2 + \text{MIS} + \text{SPU}}{\text{Required} + \text{Optional} + \text{MaximumAlternate}}$
	Maximum Error =	$\frac{\text{wrong}}{\text{Req-fills}} = \frac{\text{INC} + \text{PAR}/2 + \text{MIS} + \text{SPU}}{\text{Required} + \text{MinimumAlternate}}$

Table 4: Richness-normalized error.

For example, if system B has the raw scores in Figure 4 and if the key is filled as in Figure 5, the fill-count will range from the minimum required fills, which is a sum of Required Fills + Minimum Alternate Discretionary Fills (20+ 10), to the maximum allowed fills, which is the sum of Required Fills + Optional Discretionary Fills + Maximum Alternate Discretionary Fills (20 + 10 + 30). For this system, the richness-normalized error will range from 40/60 to 40/30 or 0.67 to 1.33.

Note that the maximum richness-normalized error can be greater than 1.00 because the fill-count in the key can be less than the number wrong for a system that overgenerates. Note also that the minimum richness-normalized error can be less than the error per response fill because the (system-independent) fill-count in the key can be greater than the (system-dependent) total used in the denominator in error per response fill.

The error score report also contains a row called "Error Rate per Word," but it should be noted that this metric is not comparable between the Japanese and the English and is not highly accurate for Japanese.

	POS	ACT	COR	PAR	INC	XCR	XPA	XIC	SPU	MIS	NON	ERR	UND	OVG	SUB
SYSTEM B			10	10	5				20	10	35				
Richness-Normalized Error						Wrong		Req-fills		All-fills		Min-err		Max-err	
						40		30		60		0.67		1.33	

Figure 4: System B.

REQUIRED FILLS	DISCRETIONARY FILLS			BLANKS
	Optional	Alternate		
		Minimum	Maximum	
20	10	10	30	35

Figure 5: Key Fills for System B.

Recall-precision Metrics

We have designated the recall, precision, and F-measure metrics that were used for MUC-4 as unofficial secondary metrics for MUC-5 in order to maintain continuity with previous MUCs. They can be used to explain current performance in comparison to past performance. Further analysis is still necessary to determine their contribution to the evaluation of data extraction systems as compared to the error-based metrics.

The recall-precision evaluation metrics were adapted from the field of Information Retrieval (IR) and extended for the MUC evaluations. They measure four different aspects of performance and an overall, combined view of performance. The four evaluation metrics of recall, precision, undergeneration, and overgeneration are calculated for the slots and in the summary score rows (see Table 5). The fifth metric, the F-measure, is a combined score for the entire system and is listed at the bottom of the report.

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]. To directly measure underpopulation or overpopulation of the template database by the information extraction systems, we introduced the measures of undergeneration and overgeneration.

recall	=	$\frac{\text{correct + (partial x 0.5)}}{\text{possible}}$
precision	=	$\frac{\text{correct + (partial x 0.5)}}{\text{actual}}$
undergeneration	=	$\frac{\text{missing}}{\text{possible}}$
overgeneration	=	$\frac{\text{spurious}}{\text{actual}}$

Table 5: Recall- Precision Evaluation Metrics.

Methods have been developed for combining the measures of recall and precision to get a single measure. In MUC-4, we used 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 useful in the context of different application requirements. 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, $\beta = 1.0$. This value is shown in the score report under the heading "P&R." The heading "2P&R" is for recall half as important as precision ($\beta = 0.5$). The heading "P&2R" is for recall twice as important as precision ($\beta = 2.0$). The F-measure is calculated from the recall and precision values in the ALL OBJECTS row.

Note that 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 what we wanted from this single measure, which we expected would encourage developers to push overall performance and, at the same time, to minimize the trade-off between the competing requirements for minimal missing, spurious, and substitution types of error.

An example showing the new metrics and the old (along with the pertinent scoring categories) for three theoretical systems is given in Figures 6 and 7. In this example, the error per response fill is the same for each of the three systems even though the F-measures are different. However, the secondary metrics of undergeneration, overgeneration, and substitution serve to distinguish the three systems. This hypothetical example points out the important role that the secondary metrics could play in system analysis as well as the analysis of the quality of the extracted information.

	POS	ACT	COR	PAR	INC	SPU	MIS	NON	ERR	UND	OVG	SUB
SYSTEM A	45	55	10	10	25	10	0	35	73	0	18	67
SYSTEM B	45	35	10	10	5	10	20	35	73	44	29	40
SYSTEM C	55	35	10	10	15	0	20	35	73	36	0	57

Figure 6: Three Systems with Equal Error per Response Fill.

	POS	ACT	COR	PAR	INC	SPU	MIS	NON	FP&R	REC	PRE
SYSTEM A	45	55	10	10	25	10	0	35	29.70	33	27
SYSTEM B	45	35	10	10	5	10	20	35	37.34	33	43
SYSTEM C	55	35	10	10	15	0	20	35	33.17	27	43

Figure 7: Unofficial Metrics for Three Systems with Equal Error per Response Fill.

Also appearing in the recall-precision score report is a row called "Text Filtering." The purpose of this row is to report how well systems distinguish relevant articles from irrelevant articles. The scoring program keeps track of how many times each of the situations in the contingency table arises for a system (see Table 6). 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 6.

The Role of the Noncommittal Scoring Category

The reader will have noticed that the category of "noncommittal" responses has been omitted from the metrics. Although this may not seem reasonable from an applications perspective, from a research perspective we believe that the exclusion of noncommittal responses results in a much less distorted cross-system view of performance. The question comes down to whether systems normally leave a slot blank out of knowledge or whether they do so out of a lack of knowledge. Highly immature systems tend either to overgenerate to an extreme, leaving few blanks, or to undergenerate to an extreme, leaving many blanks. The latter type of immature system is more common and may benefit unfairly from a metric that considers a noncommittal response to be a correct response, especially if there are relatively many blanks in the key templates.

If, for example, noncommittals were considered correct responses and included in the denominator of the error per response fill measure, the rankings of all 17 MUC-4 systems on TST3 (the name of one of the two test sets used in the evaluation) would change. The most radical changes would be for immature systems whose number of noncommittals greatly outweighs all other categories of response. Since there are a lot of immature systems evaluated for MUC-5 (as there were for MUC-4) and since the average number of fills in the answer-key templates for MUC-5 is only about half of what it was for MUC-4, the distortions of the results for MUC-5 have the potential to be even greater than they were for MUC-4. However, the potential effect on the MUC-5 evaluation is damped somewhat by the fact that the MUC-5 template consists of objects that are aligned separately; response objects that contain an insufficient amount of slot-fillers to warrant an alignment with a key object are not scored against a key object at the slot level. Nonetheless, we believe that omitting noncommittals from the metrics provides a better basis for comparison across the full range of MUC-5 (and MUC-4) systems and provides a more accurate assessment of the state of the art.

	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) **Undergeneration = c/(a+c)**

Precision = a/(a+b) **Overgeneration = b/(a+b)**

Table 6: Text Filtering.

CHANGES TO THE METRICS FROM PREVIOUS EVALUATIONS

The changes to the evaluation metrics are expected to enable three different types of evaluation "users" (NLP researchers, program managers, and potential customers) to assess and compare system performance in a meaningful way. It is also hoped that the changes will correct deficiencies in the evaluation that may unwittingly encourage conservative development strategies on the part of the researchers and that may also limit the evaluation's meaningfulness to other evaluation users.

Although the terms recall and precision were borrowed from IR, the metrics themselves represent a significant departure from the contingency table model, which underlies the IR version of the metrics. The task of extraction is a complex one that includes elements of information detection and classification, plus open-ended generation of strings and object pointers. The focus on recall and precision as primary metrics for the last few years has had some advantages, among them the following:

- they bring out the fundamental tension between spurious and missing data;
- they require that evaluation users view system performance along more than one dimension;
- they present a positive view of system performance, which may have helped to make the NLP researchers more comfortable with the idea of submitting their systems to evaluation.

However, recall and precision have the disadvantage of making a two-way distinction between error types (spurious and missing) when in fact there are three types of error. The third kind of error is captured by the substitution metric; it is accounted for by the categories of incorrect and (.5 times) partial. Substitution errors are taken into account in the recall-precision metrics to the extent that they contribute to the denominator of both recall and precision; however, this type of error is not isolated, and its inclusion in the denominator of recall and precision prevents those metrics from revealing to what extent a system's shortfalls are due to substitution rather than to missing (in the case of recall) or spurious (in the case of precision).

In a way, the recall-precision metrics view substitution as a blend of missing and spurious; a system did not simply produce the wrong fill, but rather produced a spurious fill on the one hand and missed a fill on the other hand. This is a reasonable model of system behavior in many cases, but not in others, especially when a response is scored partially correct. These deficiencies of the recall and precision metrics make the use of the error per response fill reasonable, as long as it is accompanied by the secondary metrics of overgeneration (spurious), undergeneration (missing), and substitution (incorrect, including half of the partial).

The F-measure, which was introduced for MUC-4 in response to needs of researchers and program managers for a ranking metric, has come to be used more generally than just for cross-system comparisons. By becoming the one metric of focus, it has been competing with recall and precision for the role of primary metric, thereby weakening two of the major advantages that recall and precision originally had. Furthermore, now that performance of some systems is in or approaching the 50% range, recall and precision are at a disadvantage for motivating researchers to push performance of the top systems through the more difficult stages ahead because they focus on the positive aspects of performance. These factors make the adoption of error per response fill as the primary metric a reasonable next step in determining the best way to measure performance.

The statistical significance results from MUC-5 give us feedback on how well the error metric and the F-measure distinguish systems. The results show that there are no differences between the rankings determined by error per response fill[2] and the rankings determined by F-measure. The error per response fill distinguishes systems slightly better; four more system pairs were significantly different in their error per response fill than were significantly different in their F-measure. The error per response fill also shows a tendency towards clustering systems in slightly clearer groups than the F-measure for EJV due to its ability to distinguish systems slightly better.

The richness-normalized error represents another change from previous evaluations and was motivated by the desire for a system-independent metric. The nature of this metric requires that spurious behavior be ignored. The search for such a metric led us to innovate one in which two values, a minimum and maximum, were calculated since language understanding necessarily involves variability in interpretation. It remains to be seen whether ignoring overgeneration interferes with the predictive quality of the richness-normalized error metric.

REFERENCES

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

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

[3] Nierstrasz, O. (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.

2. Although rounded numbers appear in the score report, floating point values of error per response fill were used for statistical analyses.