AN ADJUNCT TEST FOR DISCOURSE PROCESSING IN MUC-4[1]

Lynette Hirschman
Spoken Language Systems Group
MIT Laboratory for Computer Science
Cambridge, MA 02139
E-mail: hirschman@goldilocks.lcs.mit.edu

1.1 Goal of the Adjunct Test

The motivation for this adjunct test came from an exploratory study done by Beth Sundheim during MUC-3. This study showed a degradation in correctness of message processing as the information distribution in the message became more complex, that is, as slot fills were drawn from larger portions of the message and required more discourse processing to extract the information and reassemble it correctly in the required template(s). The study also suggested that systems did worse on messages requiring multiple templates than on single-template messages. These observations led us define the MUC-4 adjunct test to examine two hypotheses related to discourse complexity and expected system performance:

- The Source Complexity Hypothesis
  The more complex the distribution of the source information for filling a given slot or template (the more sentences, and the more widely separated the sentences), the more difficult it will be to process the message correctly.

- The Output Complexity Hypothesis
  The more complex the output (in terms of number of templates), the harder it will be to process the message correctly.

We began with the assumption that most systems use some variant of the following stages in creating templates:

1. Relevance filtering to weed out irrelevant portions of a message and flag relevant sentences;
2. Sentence level processing to extract information from individual units (clauses, sentences);
3. Discourse processing to establish co-reference and to merge coreferential events;
4. Template generation from the underlying sets of events, mapping events into templates.

In designing the adjunct test, our goal was to focus on the third stage, discourse processing, and to design a test that would measure differences in system performance relative to the complexity of the required discourse processing tasks. However, in complex systems such as these, it is extremely difficult to isolate one stage of processing for testing. There are many

[1]This research was supported by DARPA under Contract N00014-89-J-1332, monitored through the Office of Naval Research.

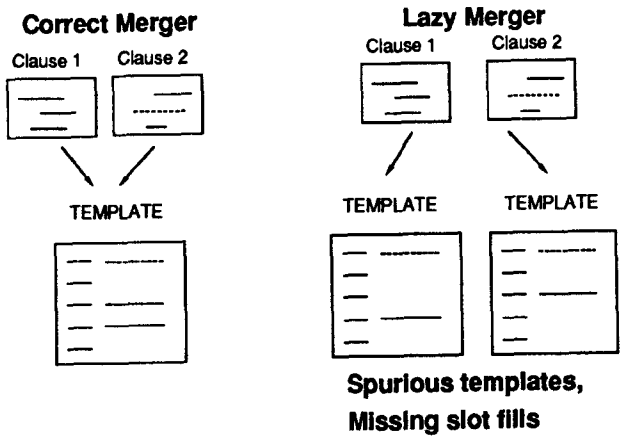

Figure 1: Lazy Merge Problem

things that can cause failure aside from discourse processing: failure to detect relevant events, failure to understand the individual sentence or clause, failure to map the information correctly into the template. Indeed, as discussed below, effects due to faulty relevance filtering masked some of the discourse issues of interest. Nonetheless, the results provide some unexpected and interesting insights into what may cause some messages to be more difficult to process than others.

1.2 To Merge or Not To Merge

In order to design a test, we focused on the event merger problem: deciding whether two clauses describe a single event or distinct events. We can distinguish two possible types of error:

- Lazy Merger
  Two clauses describe a single event and should be merged (at the template level), but the system fails to merge them (see Figure 1). This problem can occur any time a template requires more than one clause to fill the template correctly. Typically, lazy merger results in spurious templates (overgeneration at the template level); it may also result in missing slot fills.

- Greedy Merger
  Two clauses describe two different events and should not be merged. This can happen in particular when a message requires the generation of multiple templates (see Figure 2). Greedy merger typically results in missing templates and possibly in incorrect or spurious slot fills.

1.3 Experimental Design

In order to investigate problems caused by lazy merger and greedy merger, we defined two conditions: single sentence vs. multi-sentence source for a template, to test for lazy merger; and

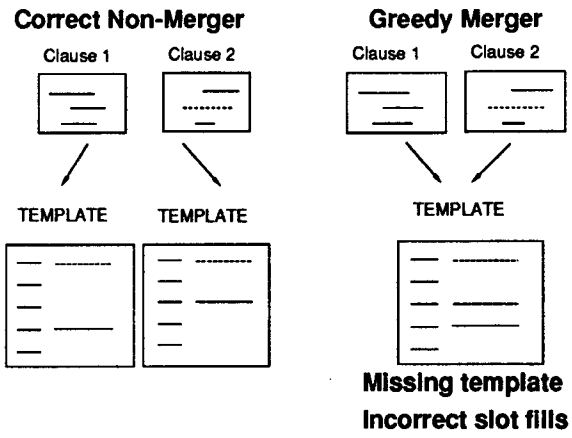

Figure 2: Greedy Merge Problem

single template vs. multi-template output, to test for greedy merger. The cross product of these conditions defines four message subsets (see Figure 3):

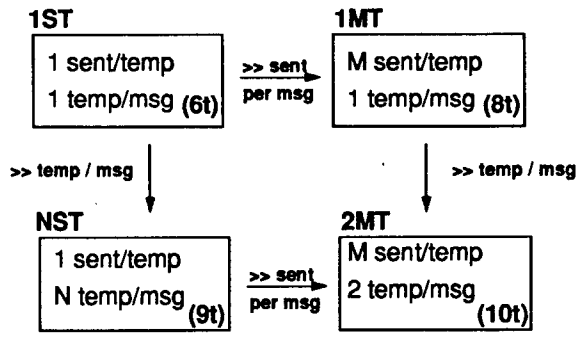

Figure 3: Test Sets

- 1ST Messages
  Generate one template, whose fill derives from a single sentence. This set would not be subject to either lazy merger or greedy merger problems.

- 1MT Messages
  Generate one template, whose fill is derived from more than one sentence. This set would be subject to lazy merger problems, but not greedy merger problems.

- NST Messages
  Generate multiple templates, but each template is derived from only a single sentence. These would be subject to greedy merger problems, but not lazy merger problems.

- 2MT Messages
  Generate two templates, each requiring multiple sentences to fill. These messages should be the hardest set, since they will be subject to both lazy merger and greedy merger problems. They should show lazy merger problems relative to the NST set and greedy merger problems relative to the 1MT set.

We then examined the TST3 message set and found messages to populate each subset. The adjunct test thus required four separate scoring runs, one for each subset. A total of 23 messages were involved, 4–8 messages and 6–10 templates per subset (see Appendix 1 for the test set composition). Messages containing optional templates were rejected[2], and of course irrelevant messages did not fit into any test set. In general, messages that were "mixed" also did not fit into any subset.

Unfortunately, it turned out there were problems with this methodology. The first problem was that there were few instances of templates meeting these specifications, other than the 1MT set. In particular, there were few multi-template messages where all templates were derived from only a single sentence (the NST set). To try to preserve this set, we compromised by scoring just those templates within each message that were generated from single sentences, which in turn meant that we could not use the MATCHED-SPURIOUS or ALL-TEMPLATE measures, since these require scoring *all* of the templates associated with a given message.

The second problem had to do with the single-sentence, single-template messages (the 1ST set). It turned out that these messages were rare, and quite different in character from the more common 1MT messages which generated a template from multiple sentences. Clearly, the 1ST subset posed a particularly hard problem in terms of relevance filtering – how to process the one relevant sentence in the message, in the face of the "noise" of the rest of the message. For this reason, the results on 1ST turned out to be more about relevance filtering than about discourse processing. This is discussed in more detail below.

1.4 Measuring Lazy Merger and Greedy Merger

Using these four message subsets, we then asked how lazy merger and greedy merger would affect the various scores reported by the scoring program. The effects included both slot-level effects (missing slot fills, incorrect or spurious slot fills within the expected template), and template level effects (spurious templates, missing templates). Slot-level effects could be measured in terms of the MATCHED-ONLY calculations. Missing templates could be measured using the MATCHED-MISSING (or ALL-TEMPLATES) metrics, and spurious templates in terms of the MATCHED-SPURIOUS (or ALL-TEMPLATES) metrics.

We expected lazy merger to produce extra templates, measured as overgeneration in the MATCHED-SPURIOUS metric[3]. Lazy merger also should lead to missing slot fills, where information from the second event should have been folded into the template, but instead led

[2]Except for message 48 in the 2MT set, which, by oversight, had an optional template.

[3]Or perhaps more accurately, as the difference between MATCHED-SPURIOUS or ALL-TEMPLATES overgeneration minus MATCHED-ONLY overgeneration. Since MATCHED-ONLY overgeneration measures slot level overgeneration, the difference would separate out only the template level overgeneration. However, in the measurements below, the ALL-TEMPLATE metric alone was used.

to generation of a new template. This could be measured by slot level undergeneration, defined as *Missing/Possible* using the MATCHED-ONLY metric[4].

Since lazy merger problems arise when multiple clauses/sentences contain information, redundancy might offset some of these problems. If the same piece of information were to occur in several places, this would increase the probability of recall on that slot. This might also have an affect on precision, by increasing the number of correctly filled slots, relative to those filled incorrectly.

Greedy merger could result in lower recall at the template level, because it would produce too few templates, each with too much information in it (spurious or incorrect fills). The missing templates would cause undergeneration, namely a lower ratio of filled slots to possible slots in the MATCHED-MISSING or ALL-TEMPLATES measures, and a corresponding decrease in recall. Greedy merger could also result in *incorrect fills*, when fills from two clauses are incorrectly combined in a single slot. This could be measured by the number incorrect slot fills over number of actual fills in the MATCHED-ONLY data.

Failure to filter *irrelevant* clauses could affect all the results by providing additional events which could be made into (spurious) templates or merged incorrectly. Spurious templates cause overgeneration and loss of precision (measured in MATCHED-SPURIOUS or ALL-TEMPLATES)[5], and, incorrect merger of events can cause spurious or incorrect slot fills (lower precision and possibly lower recall in MATCHED-ONLY).

Figure 4 illustrates the relation of the four test subsets, and the hypothesized findings. Note that we compare sets 1ST vs. 1MT and NST vs. 2MT for issues of lazy merger; and sets 1ST vs. NST and 1MT vs. 2MT for greedy merger. Finally, we expect 1ST to show higher precision and recall (higher F-score) than 2MT.

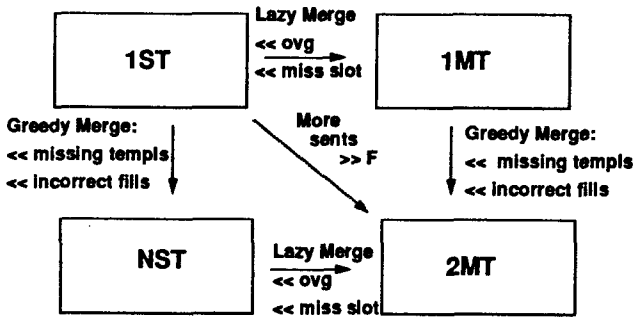

Figure 4: Hypothesized Results

[4]This could cause loss of recall, because of the increase in partially filled templates and loss of precision (in the MATCHED-SPURIOUS or ALL-TEMPLATES measure), due to spurious templates.

[5]But they have no affect on the MATCHED-ONLY measures.

Measure: ALL TEMPLATES	1ST	1MT	2MT
Overgeneration - All Systems	77	48	56
Overgeneration - Top 8 Systems	57	33	35

Table 1: Overgeneration of Templates

1.5 Results

1.5.1 Lazy Merger Results

As discussed in the preceding section, we expected the single-sentence messages to show less template overgeneration than the multi-sentence messages (1ST vs. 1MT and NST vs. 2MT). However, exactly the opposite occurred: the median overgeneration score (ALL-TEMPLATES, all systems) for 1ST was 77%, compared to 48% for 1MT (and, though not directly comparable, 56% for 2MT)[6]. These relative results held for the top 8 systems as well. These results are shown in Figure 5; the stripe indicates the median, the dark region is encompasses the middle two quartiles, and the brackets indicate the range of the data. Outliers are plotted as additional lines. The overall results are summarized in Table 1. We conclude that problems in relevance filtering for the 1ST messages vastly overshadowed any affect of lazy merger problems.

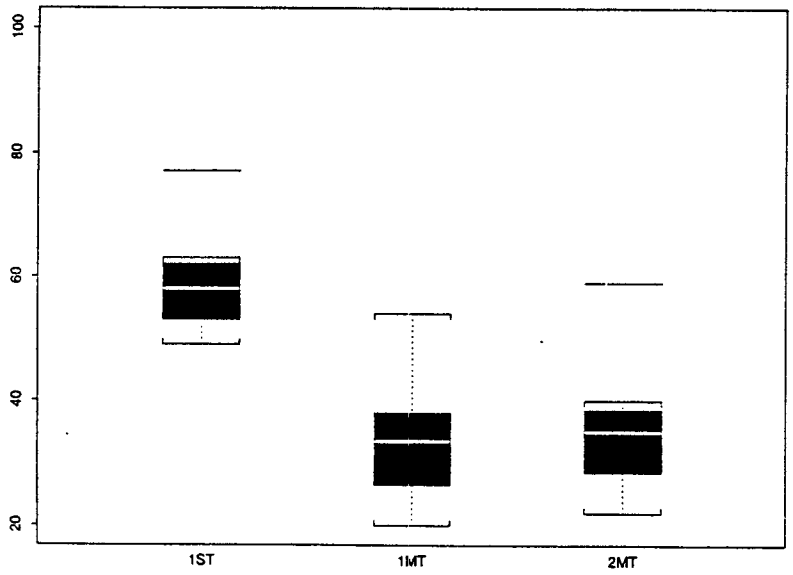

Figure 5: Lazy Merger: Overgeneration Results on Top 8 Systems

The other hypothesis associated with lazy merger was missing slots fills, measured on the

[6]Note that we could not include the overgeneration result for the NST set, because these values were measured on partial messages, invalidating all scores other than MATCHED-ONLY.

Measure: MATCHED ONLY	1ST	1MT	NST	2MT
Undergeneration	14	24	11	33
Ave. No. Possible Fills/Template	5.5	9.5	3.7	9.9

Table 2: Undergeneration of Slot Fills

Measure: ALL TEMPLATES	1ST	1MT	2MT
Undergeneration - All Systems	54	51	59
Undergeneration - Top 8 Systems	40	38	49
Possible No. Slot Fills/Template	10.4	14.4	16.2

Table 3: Undergeneration of Templates

MATCHED-ONLY data (which allows us to use all four test subsets). Table 2 shows "undergeneration" for these four test sets, where undergeneration is defined as *Missing/Possible.*

In this case, the results are consistent with our hypothesis of lazy merger. However, it turns out that they are equally consistent with another hypothesis, namely that the number of missing slots fills will be correlated with the number of possible slots per template. Since templates generated from a single clause are typically much more sparse than templates generated from multiple clauses, this appears to be at least as good an explanation of the observed results. The second row of Table 2 shows the average number of slot fills for each class. Note that NST has the lowest undergeneration score, and the fewest slot fills, followed by 1ST, followed by 1MT and finally 2MT.

1.5.2 Greedy Merger Results

For greedy merger, we hypothesized that multi-template messages would show more missing templates, as well as more spurious and incorrect slot fills (comparing 1ST to NST and 1MT to 2MT). Again, the NST test subset could not be used in looking at spurious templates. Comparing 1MT to 2MT, the results were as expected: 1MT had 51% undergeneration (*Missing/Possible* using the ALL-TEMPLATES figures), and 2MT had 59%, averaged over all of the systems; the difference was more pronounced for the top 8 systems (1MT undergeneration was 38%, 2MT was 49%). The 1ST results were 54% (40% for the top 8 systems), higher than 1MT, perhaps due to losing some templates because of faulty relevance filtering. These figures are shown in Table 3.

The second prediction about greedy merger concerned incorrect slot fills, resulting from combining fills from two different clauses. This was calculated by dividing the number of incorrect fills over the number of actual fills, for the MATCHED-ONLY measure. Here the results were negative. The average over all systems showed 1ST equal to NST and 1MT greater than 2MT. For the top 8 systems, the difference between 1MT and 2MT disappeared as well. The dom-

Measure: MATCHED ONLY	1ST	NST	1MT	2MT
Incorrect/Actual - All Systems	6	6	17	12
Incorrect/Actual - Top 8 Systems	5	5	12	11
Ave. Actual Slots/Template	5	4	10	10

Table 4: Incorrect Slot Fills in MATCHED ONLY Measure

inant affect was that the multi-sentence per template sets (1MT, 2MT) had more than twice the number incorrect compared to the single-sentence per template sets (1ST, NST); the figures are given in Table 4. It is unclear how to interpret these results, except to note that there were twice as many fills generated for the 1MT and 2MT sets (10 per template, on average), as for the 1ST and NST sets (around 5 fills/template).

Finally, we predicted that the 1ST subset would be the easiest, and the 2MT set the hardest overall, measured in terms of the F-score. Here, the affects of the poor performance on the 1ST set were quite striking. For example, Figure 6 shows a plot the F-score for 1ST vs. F-score for the whole of TST3. Only 3 systems (Hughes, BBN, NYU) did better on 1ST than on TST3 as a whole.

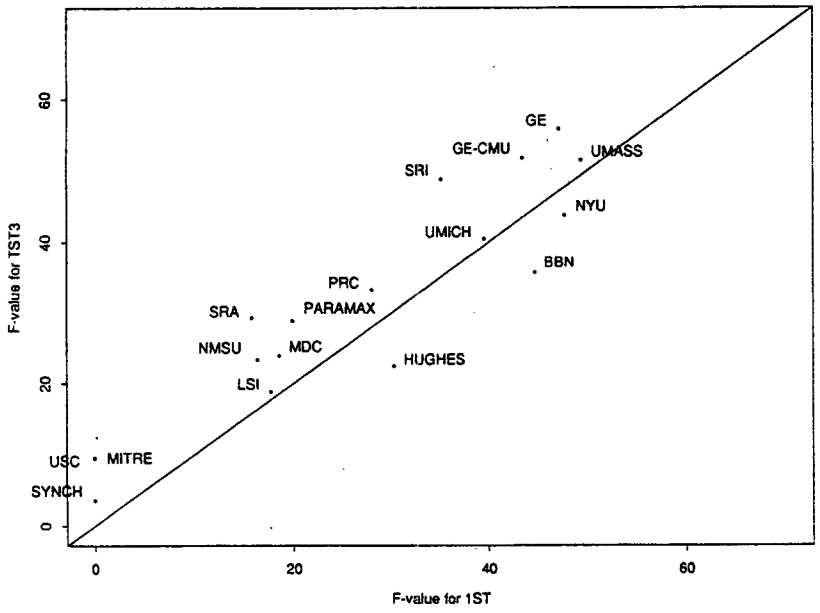

Figure 6: F-Scores for the 1ST Set vs. Overall F-Scores for TST3

On the other hand, if we plot F-scores for 1MT against F-scores for TST3, the distribution is much more even (see Figure 7). In general, most systems scored substantially better on the 1MT set (39% F-score on ALL-TEMPLATES) than on the 1ST set (28%), contrary to the predictions. However, the score on 1MT was higher than the score on 2MT, as predicted (39%

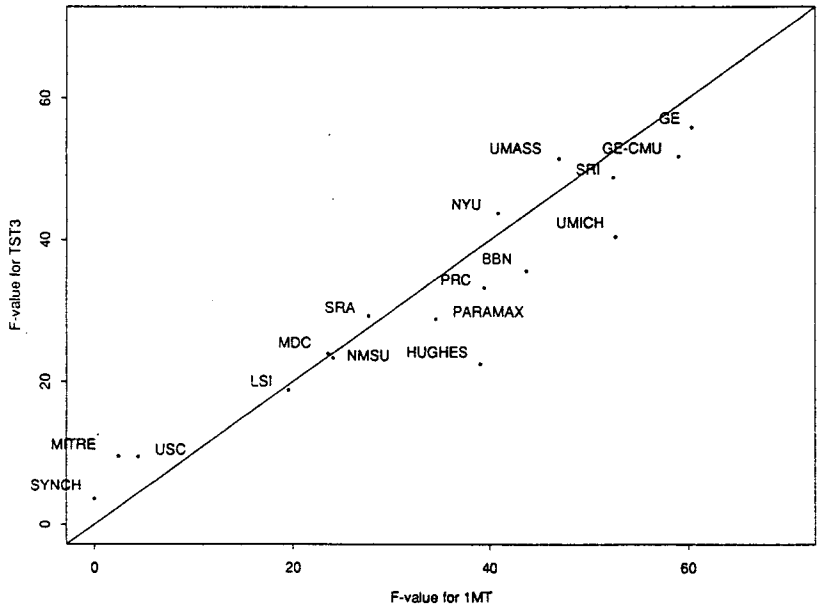

Figure 7: F-Scores for the 1MT Set vs. Overall F-Scores for TST3

Measure	1ST	1MT	2MT
F-scores - All Systems	28	39	29
F-scores - Top 8 Systems	44	50	48

Table 5: ALL TEMPLATE F-Scores

vs. 29%). There was a somewhat smaller effect for the top 8 systems, shown in Table 5 below.

Figure 8 shows graphically the relationship of the ALL-TEMPLATES F-score for the top 8 systems. Five of the eight systems do much better on 1MT, while the other three systems do slightly worse.

The overall results of these tests are summarized in Figure 9.

1.6 Conclusions

We can draw several conclusions from this experiment. First, the 1ST message subset turned out to be quite anomalous. It was harder than the 1MT set, as seen in the F-scores, as well as in the overgeneration results. This is most likely attributable to a relevance filtering problem. The 1ST messages were peculiar in that the the single relevant sentence was embedded in a message that was generally focused on something else; the relevant event was only mentioned as background, or in passing. Understandably, the systems had trouble picking out the one relevant sentence amidst a text of otherwise irrelevant information.

The second finding is that the 2MT subset was indeed harder than the 1MT set; six out

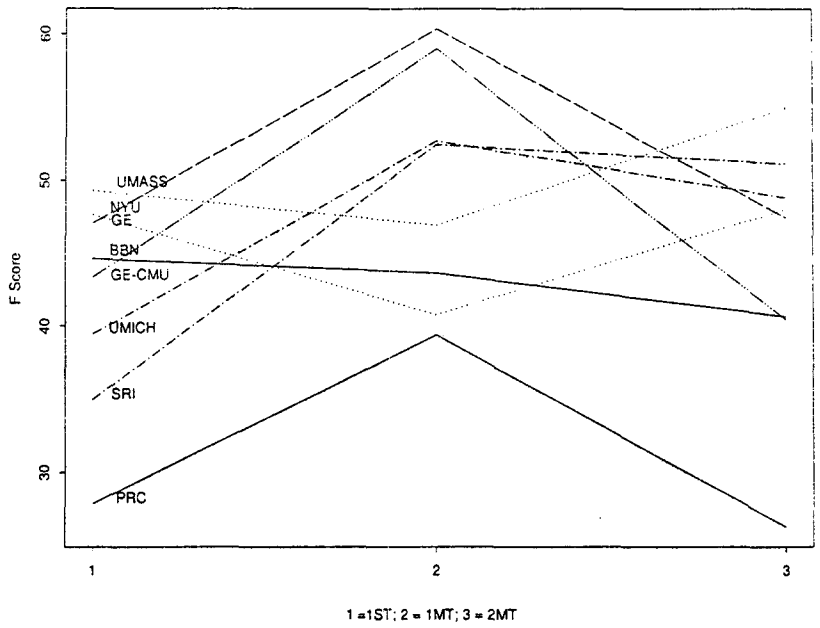

Figure 8: ALL TEMPLATE F-Scores for the Top 8 Systems on Sets 1ST, 1MT, 2MT

of the 8 top systems did worse on 2MT than on 1MT, as measured by the ALL-TEMPLATES F-score. It seems possible that at least some of this may be due to greedy merger problems, supported by the somewhat greater template undergeneration for 2MT relative to 1MT.

Next, a surprising result was the relative consistency of the behavior of the various systems with respect to the message subsets. In general, most results held regardless of whether the results were obtained by averaging across all systems, or over just the top 8 systems. Given the enormous variation in system maturity and performance, this is quite surprising, and leads to the hypothesis that some messages may simply be harder than others, across all systems.

Finally, at least anecdotally, many systems reported instances of both these problems. It may be that the affects of these discourse level problems were masked at times by other problems (relevance filtering, for example). Nonetheless, we can conclude that lazy merger and greedy merger are real problems in discourse processing.

The results of this test suggest several further research directions and possible future adjunct tests. First, the problem of distinguishing between relevant and irrelevant information caused significant performance degradation, as evidenced by the difference between F-scores for MATCHED-ONLY and F-scores for ALL-TEMPLATES. This should be investigated further, possibly by looking at system performance on the irrelevant messages as well.

Second, it may be worth investigating some measure of the relative difficulty across messages, for example, by computing performance statistics across messages rather than across systems. We would expect to see significant variation in these scores, and this might lead us to understand better what constitutes a hard message. Apparently, subset 1ST constituted such a set.

Third, this paper analyzed the results averaged over systems, with no attempt to compare individual systems. The question remains as to whether these measures will provide some useful

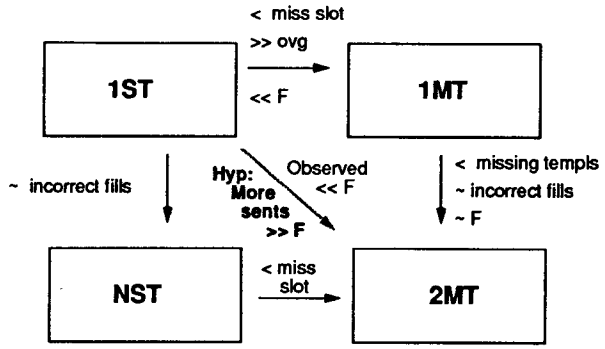

Figure 9: Hypothesized Results

diagnostics or insights to individual system developers, although that investigation was beyond the scope of this paper.

In conclusion, this adjunct test was admittedly crude, with too few messages and many uncontrolled variables. Nonetheless, the test provided new and unexpected insights into some variables affecting system performance. In addition, the adjunct test methodology adopted here is of interest because the test was carried out simply by rescoring various subsets of the original test – thus avoiding the need to conduct a separate test, with different input. Also, it was primarily a "within system" test – that is, each system was compared to itself, rather than to other sites. For these reasons, this methodology is worth exploring in the design of future adjunct tests.

1.7 ACKNOWLEDGEMENTS

I would like to acknowledge the assistance received from Beth Sundheim and Nancy Chinchor for providing feedback and guidance in defining the adjunct test sets and for preparation of the scoring runs.

1.8 APPENDIX: THE TEST SETS

- 1ST (6 messages, 6 templates) 19, 33, 66 74, 82, 98
- 1MT (8 messages, 8 templates) 3, 5, 20, 27, 34, 44, 73, 91
- NST (4 messages, 9 templates) 38[1,2,3], 24[3,4], 30[3,6], 94[3,6]
- 2MT (5 messages, 10 templates) 37, 40, 48[1,2] 50, 84