APPENDIX B:

PROCEDURE FOR MUC-4 FINAL TESTING

NOTE: This test procedure references the following files to be ftp'ed from /pub/muctest. READ THE TEST PROCEDURE BEFORE ACCESSING THESE FILES.

tst3-muc4
tst4-muc4
key-tst3.v1
key-tst4.v1
slotconfig-tst3.el
slotconfig-tst4.el
README-adjunct-test1

config-tst3.el
config-tst4.el
config-progress-tst3.el
config-1ST-tst3.el
config-1MT-tst3.el
config-NST-tst3.el
config-2MT-tst3.el

1. SCHEDULE

You are not to ftp the test files until you are ready to start testing. Testing may be done any time between 26-31 May. The only requirement is that all reports (see section 7, below) be submitted by first thing Monday morning, 1 June. Permission to attend MUC-4 in June may be revoked if you do not meet this deadline!

If you intend to carry out any of the optional testing (see below, section 4), you must report the planned optional test(s) to NRaD before starting the test procedure. This means that you should describe, in some meaningful terms, specifically how you will alter the behavior of the system and what kind of performance differences you expect to obtain.

2. PERFORMANCE OBJECTIVES

In reporting the results of MUC-4, we will be focusing on three aspects of the scoring:

a. Recall and precision in the Matched Only (MO), Matched/Missing (M/M), Matched/Spurious (M/S), and All Templates (AT) rows. When displayed together in a scatter plot, the four data points form the corners of a rectangle that we are calling a system's basic "region of performance."

b. The overgeneration scores in the MO, M/M, M/S, and AT rows.

c. The recall and precision scores in the Text Filtering row.

When it is necessary to single out one set of scores from among the MO, M/M, and M/S, and AT rows, we will usually single out the AT scores, since they penalize equally for missing and spurious data (unlike M/M and M/S) and they penalize both at the template level and at the slot-filler level (unlike MO). Statistical significance testing of the overall results will be done on the basis of the AT scores.

When it is advisable to present a scientifically valid means of determining a ranking of the systems, we will use a formula for calculating what is known as the F-measure. Given two systems whose overall recall and precision sum up to the same number and given equal weights for recall and precision in the F-measure formula, the formula will rank the system whose recall and precision are more equal higher than the system whose recall and precision are more divergent. In order to show how the rankings may vary depending on the relative weight assigned to recall vs precision, we will present three different calculations of the F-measure, one in which recall and precision are weighted equally, one in which recall is weighted twice as heavily as precision, and one in which precision is weighted twice as heavily as recall. We intend to conduct statistical significance testing at least for the version of the formula in which the weight of recall is equal to the weight of precision.

3. REQUIRED TESTING

The final test has three required components:

a. a template-by-template and message-by-message performance test on TST3, which is a test set of 100 articles taken from the same source and covering the same time period as those that comprise DEV, TST1, and TST2;

b. a template-by-template and message-by-message performance test on TST4, which is a test set of 100 articles taken from the same source as the other sets but representing incidents from a somewhat different time period;

c. an "adjunct" performance test on TST3 in which selected messages in the test set have been sorted into different categories and are to be scored separately template by template. See the README-adjunct-test1 file for further information on the nature of this test.

[A second adjunct test will be carried out by GE and will require no additional effort on the part of the other participants. A description of this adjunct test is provided in README-adjunct-test2. Please note that if you do not wish to participate in this test, you must notify NRaD by June 1.]

To complete the required testing, you will need approximately the same amount of time as it would normally take the system to produce templates for two sets of 100 new texts and for you to interactively score them once (to do any manual remappings and produce template-by-template score reports) and to non-interactively make several more runs (a) to produce message-by-message score reports (total of 2 scoring runs) and (b) to use the remaining configuration files to produce template-by-template score reports for the adjunct test (total of 4 short scoring runs) and for measuring progress since MUC-3 (1 scoring run).

4. OPTIONAL TESTING

You are encouraged to design interesting experiments in which you hypothesize significant performance differences that can be obtained by such means as removing a module or inserting one that's not part of the basis system or by altering the control structure of the system such that it produces templates more aggressively or more conservatively. An experiment may result in a single set of new scores or in a continuous recall-precision "curve".

The objective of the optional testing is to learn more about the controlled tradeoffs that some systems may be designed to make between recall and precision. If your system meets one of the following two criteria, it is a candidate for optional testing:

a) if the system can control the tradeoff between recall and precision in order to produce a set of data points sufficient to plot the outline of a recall-precision curve;

b) if the system's recall and precision can be consciously manipulated by the loosening or tightening of analysis constraints, etc., in order to produce at least one data point that contrasts in an interesting way with the results produced by the required testing.

5. TEST PROCEDURE

5.1 Freezing the System and FTP'ing the Test Package

When you are ready to run the test, ftp the files in the test package from /pub/muctest. You are on your honor not to do this until you have completely frozen your system and are ready to conduct the test. You must stop all system development once you have ftp'ed the test package.

Note: If you expect to be running the test over the weekend and are concerned that a host or network problem might interfere with your ability to ftp, you may ftp the files on Friday. However, for your own sake, minimize the accessibility of those files, e.g., put them in a protected directory of someone who is not directly involved in system development.

5.2 Generating the System Response Templates

There are 100 texts in tst3-muc4 and 100 texts in tst4-muc4. Without looking at the texts, run your system against the files and name the output files response.tst3 and response.tst4, respectively. (For your information, the format of the message IDs is TST3-MUC4-nnnn and TST4-MUC4-nnnn.)

You are to run the test only once -- you are not permitted to make any changes to your system until the test is completed. If you get part way through the test and get an error that requires user intervention, you may intervene only to the extent that you are able to continue processing with the NEXT message. You are not allowed to back up!

Notes:

1) If you run short on time and wish to break up the test sets and run portions of them in parallel, that's fine as long as you are truly running in parallel with a single system or can completely simulate a parallel environment, i.e., the systems are identically configured. You must also be sure to concatenate the outputs before submitting them to the scoring program.

2) No debugging of linguistic capability can be done when the system breaks. For example, if your system breaks when it encounters an unknown word and your only option for a graceful recovery is to define the word, then abort processing and start it up again on the next test message.

3) If you get an error that requires that you reboot the system, you may do so, but you must pick up processing with the message FOLLOWING the one that was being processed when the error occurred. If, in order to pick up processing at that point, you need to create a new version of the test set that excludes the messages already processed or you need to start a new output file, that's ok. Be sure to concatenate the output files before submitting them to the scoring program.

5.3 Editing Config Files to Supply Proper Pathnames

Follow the instructions in this section before initializing the scoring program.

The scoring program configuration (config) files contain arguments to the define-muc-configuration-options function, which you will have to edit to supply the proper pathnames. Make no further edits to the config files.

Also included in the test package are slotconfig-tst3.el and slotconfig-tst4.el, which have been updated to recognize the message IDs that are used in the test sets. Be sure that you have put the right pathname to each slotconfig file in each config file.

5.4 Remapping Templates

It is recommended that this step be carried out BEFORE you start scoring.

After the scoring program has been initialized using config-tst3.el or config-tst4.el (or config files for any optional tests) as the argument to initialize-muc-scorer, you may wish to browse through the templates to see if there are any mappings you wish to change using the manual template remapping feature of the scoring program. When you have finished updating the mappings, exit the browser and continue with the instructions given below.

5.5 Scoring the System Response Templates

Follow the instructions in this section each time the scoring program is initialized.

5.5.1 For the Basic Test

Scoring for the basic test is done by using config-tst3.el, config-tst4.el, and config-progress-tst3.el.

5.5.1.1 Template-by-Template Scoring

This section applies to config-tst3.el and config-tst4.el.

Having started up the scoring program using config-tst3.el or config-tst4.el as the argument to initialize-muc-scorer, type C-u s (i.e., Control-u followed by the letter "s") so that the scoring program will produce template-by-template score reports.

Refer to the interactive scoring guidelines while scoring. When you have finished scoring, save the score buffer (*MUC Score Display*) to the appropriate file name:

a) for config-tst3.el, save the score buffer to scores.tst3-pass1;
b) for config-tst4.el, save it to scores.tst4-pass1.

After saving the score buffer, save the history to file using the "h" command (the config file specifies the history file name).

5.5.1.2 Message-by-Message Scoring

This section applies to config-tst3.el and config-tst4.el.

Before you reinitialize the scoring program with the next config file, type C-u l (i.e., Control-u followed by the letter "l") so that the scoring program will do a message-by-message scoring and the final summary table in the score buffer will include the TEXT FILTERING row.

When you have finished scoring, save the score buffer (*MUC Score Display*) to the appropriate file name:

a) for config-tst3.el, save it to scores.tst3-pass2;
b) for config-tst4.el, save it to scores.tst4-pass2.

After saving the score buffer, save the history to file using the "h" command (this will overwrite the version saved at the end of the template-by-template run).

5.5.1.3 Scoring to Measure Progress Since MUC-3

This section applies only to config-progress-tst3.el.

This scoring run is to be done only for TST3; it makes use of config-progress-tst3.el. The results of this scoring will be used as a point of comparison with MUC-3. Therefore, the "display-type" option is set to "matched-missing" rather than to "all-templates." This run scores only the slots that are not in conflict with the template design that was used last year for MUC-3. This means that it does not score the instrument slots nor any of the number slots.

Even if you did not participate in MUC-3, you are asked to make this scoring run. (NRaD is using a similar config file to rescore an updated version of last year's response templates for MUC-3 veteran sites.)

The config file specifies that this scoring run will make use of the history file that you created when you originally scored TST3. Thus, no interaction should be needed when scoring.

Having started up the scoring program using config-progress-tst3.el as the argument to initialize-muc-scorer, type C-u s (i.e., Control-u followed by the letter "s") so that the scoring program will produce template-by-template score reports.

When you have finished scoring, save the score buffer (*MUC Score Display*) to scores-progress.tst3.

You do not need to save the history file, and you do not need to do message-by-message scoring.

5.5.2 For the Adjunct Test Described in "README-adjunct-test1"

Adjunct test scoring makes use of the following config files:

a) config-1ST-tst3.el,
b) config-1MT-tst3.el,
c) config-NST-tst3.el,
d) config-2MT-tst3.el.

Each config file specifies that the scoring run will make use of the history file that you created when you originally scored TST3. Thus, no interaction should be needed when scoring. Furthermore, each run will score only a small number of templates; thus, it should take little time to complete each run.

Having started up the scoring program using the appropriate config file as the argument to initialize-muc-scorer, type C-u s (i.e., Control-u followed by the letter "s") so that the scoring program will produce template-by-template score reports.

When you have finished scoring, save the score buffer (*MUC Score Display*) to the appropriate file name:

a) for config-1ST-tst3.el, save it to scores-1ST.tst3;
b) for config-1MT-tst3.el, save it to scores-1MT.tst3;
c) for config-NST-tst3.el, save it to scores-NST.tst3;
d) for config-2MT-tst3.el, save it to scores-2MT.tst3.

You do not need to save the history file, and you do not need to do message-by-message scoring.

6. SPECIAL INSTRUCTIONS FOR OPTIONAL TESTING

For each optional run, modify the system as you described in advance to NRaD. Then follow the applicable procedures in section 5 to produce and score new templates for TST3, using modified versions of config-tst3.el. Depending on the objectives of your optional testing, you should produce template-by-template scores, message-by-message scores, or both.

To yield these additional data points, you will generate and score new system response templates for TST3, using the history file generated during the required testing. NO SYSTEM DEVELOPMENT IS PERMITTED BETWEEN OFFICIAL TESTING AND OPTIONAL TESTING -- ONLY MODIFICATION OF SYSTEM CONTROL PARAMETERS AND/OR REINSERTION OR DELETION OF EXISTING CODE THAT AFFECTS THE SYSTEM'S BEHAVIOR WITH RESPECT TO THE TRADEOFF BETWEEN RECALL AND PRECISION.

If, as a consequence of altering the system's behavior, templates are generated that weren't generated during the required testing or slots are filled differently, you may find it necessary to add to the history file and to change some of the manual template remappings. Start the scoring of each optional test with the history file generated during the previous run, minus the manual template remappings; save any updated histories to new file names.

In order to obtain these data points, you may wish to conduct a number of tests and throw out all but the best ones. Remember, however, that you are to notify NRaD of ALL the planned experiments in advance (see section 1). Thus, it would be wise to experiment on the training data and use the results to know what different runs are worth making during the test.

If you wish to conduct your optional tests on TST4 as well as on TST3, you may do so, but please submit the results for TST4 only if you find the differences in scores to be significant.

Once you have determined which of the optional runs to submit to NRaD for the official record, name the files for those runs in some meaningful, easily-understood fashion, fitting these patterns:

a) response-<meaningful-name-here>.tst3,
b) history-<meaningful-name-here>.tst3,
c) scores-<meaningful-name-here>.tst3-pass1,
OR scores-<meaningful-name-here>.tst3-pass2,
OR BOTH OF THE ABOVE,
d) trace-<meaningful-name-here>.tst3.

7. REPORTS TO BE SUBMITTED BY MONDAY MORNING, JUNE 1

7.1 List of Expected Files

Following is a summary list of the expected files:

1. response.tst3
   response.tst4
2. history.tst3
   history.tst4
3. scores.tst3-pass1 (template-by-template scores)
   scores.tst3-pass2 (message-by-message scores)
   scores.tst4-pass1 (template-by-template scores)
   scores.tst4-pass2 (message-by-message scores)
   scores-progress.tst3
   scores-1ST.tst3
   scores-1MT.tst3
   scores-NST.tst3
   scores-2MT.tst3
4. trace-tst3 (system trace for the 100 TST3 messages)
   trace-tst4 (system trace for the 100 TST4 messages)
   -- You may submit whatever you think is appropriate, i.e., whatever would serve to help validate the results of testing.

Additional response, history, score, and trace files are expected for each optional test that you wish to have included in the official record.

7.2 How to Submit Files

Before you submit the expected files, PLEASE TAR AND COMPRESS THE FILES. Please help us identify your files by labeling the compressed tar file as follows: <site-name>-muctest.TAR.Z.

The compressed tar file is to be submitted via anonymous ftp to the directory named /incoming. Please notify NRaD and SAIC by email after the file has been successfully transferred.

8.0 RESCORING OF RESULTS

The interactive scoring should be done in strict conformance to the scoring guidelines. If you perceive errors or other problems in the guidelines or in the answer keys as you are doing the scoring, please make note of them and send a summary to NRaD.

We will rescore everyone's response-muc4.tst3 files, creating a cumulative history file. Your notes on perceived errors will be useful to us when we prepare to do that rescoring. We will then distribute the cumulative history file to all sites and will send the individual sites our version of their system's complete score report. The rescored version of the summary score reports will be labeled anonymously and distributed to the MUC-4 sites prior to the conference.