UNL/USL: MUC-3 TEST RESULTS AND ANALYSIS

Jitender S. Deogun
Department of Computer Science & Engineering
University of Nebraska - Lincoln
Lincoln, NE 68588-0115
deogun@fergvax.unl.edu
(402)-472-5033

Vijay V. Raghavan
Center for Advanced Computer Studies
University of Southwestern Louisiana
Lafayette, LA 70504-4330
raghavan@cacs.usl.edu
(318)-231-6603

RESULTS

This report describes the results from the required run in addition to the five optional runs for the MUC-3 task. One purpose of the optional runs is to investigate the precision-recall tradeoff. Two of the optional runs (options 1 and 4) were also submitted to NOSC, making up a total of three official runs.

The experiments are conducted using three different training sets selected from the corpus of 1300 messages for which the key-templates have been manually generated during the first phase of the MUC-3 project. The Training Set 1 (200 messages) and the Training Set 2 (300 messages) contain an almost equal number of relevant and non-relevant messages where a message is termed relevant if it generates at least one template. The Training Set 1 is a proper subset of Training Set 2. The Training Set 3 (306 messages) contains only those messages that are relevant and generate *one and only one* template. The occurrence distribution of the various incident types for the three training sets are presented in Table 1. The *ed rows indicate which of the incident types have enough occurrences in the training set to be learnable. Similarly, not all fills associated with other slots are learnable.

In any run, only two of the three training sets are used. One of these two training sets is used to develop a rule vector (termed as the *optimal_query*) that can identify a message as being relevant to the MUC-3 task. The other training set is used to develop concept rule vectors that can identify which among the various possible slot fills are actually applicable[1] to a message. Since our system mainly deals with slots for which fills come from a predefined set of fills (i.e. these are identified with concepts to be learned), the number of concept rule vectors that pertain to each slot is not too many.

The activation value for a slot fill with respect to a test message is computed as the dot product of the concept rule vector and the message representation (as a vector). For the required and some of the optional runs (options 2, 4, and 5), the system decides that a slot fill applies if its activation value with respect to the rule vector for the slot fill is greater than a dynamically generated threshold T1. This threshold for a given slot fill is based on the percentage of messages in the training set to which the set fill is applicable and the histogram depicting the distribution of the activation values

[1] "applicable" means relevance of fill to a message

Incident Type	Training Set 1	Training Set 2	Training Set 3	
ARSON	10	11	5	*
ARSON THREAT	0	1	0	
MURDER	59	79	136	*
DEATH THREAT	9	12	8	*
BOMBING	42	63	71	*
BOMB THREAT	0	1	1	
KIDNAPPING	15	17	25	*
KIDNAPPING THREAT	0	0	0	
HIJACKING	0	0	0	
HIJACKING THREAT	0	0	0	
ROBBERY	4	6	3	
ROBBERY THREAT	0	0	0	
ATTACK	23	43	44	*
ATTEMPTED ARSON	0	0	0	
ATTEMPTED MURDER	1	5	7	
ATTEMPTED BOMBING	9	12	6	*
ATTEMPTED KIDNAPPING	0	0	0	
ATTEMPTED HIJACKING	0	0	0	
ATTEMPTED ROBBERY	0	0	0	
Total relevant	124	172	306	
Total nonrelevant	76	128	0	
* incident types for which concept rule vectors are generated by the learning module.				

Table 1: Frequency of incident types in training sets

Test Run	Optimal Query**	Set-list type fills or Concepts***	Threshold	Comment
Required	Training Set 1	Training Set 2	T1	
Option 1	Training Set 1	Training Set 2	0	
Option 2	Training Set 2	Training Set 1	T1	
Option 3	Training Set 2	Training Set 1	0	
Option 4	Training Set 2	Training Set 3	T1	
Option 5	Training Set 2	Training Set 3	T1	No phrases
** Optimal query needs non-relevant messages in the training set.				
*** When non-relevant messages are present in the training set for concepts, they are treated as negative examples for every concept.				

Table 2: Different parameter settings

in the test set of messages for this slot fill. A second option is to use a zero threshold implying that the slot fill is applicable if the activation value of the corresponding rule vector with respect to the message representation is positive.

The training sets and the threshold setting used for official and optional test runs are presented in Table 2. The number of templates generated are compared in Table 3 and detailed results in terms of precision, recall, and overgeneration for Option 4 are presented in Table 4. Since our system placed an emphasis on set list type slot fills, our system's performance, with respect to **set fills only**, for the various test runs, is summarized in Table 5. The results for Options 2, 3, and 5, shown in Tables 3 and 5, are scored at our site rather than by official scorers. Consequently, these figures are not completely consistent with those of the other options. In our assessment, the recall and precision values of our scoring are lower in Table 3 than what they would have been if scored by the official scorers. In contrast, the same options in Table 5 are somewhat inflated compared to what the official scoring would have yielded. With this disparity in mind, the following observations are made on results from different runs:

Required Run (Official-1) The official run generated a large number of templates. This run resulted in a moderate recall and moderate precision.

Option 1 (Official-2) The run for option 1 does not generate many templates. This option uses a stricter or higher threshold for concepts compared to the Official-1 run. Therefore, this method achieves a low recall with a reasonable level of precision. This option sacrifices recall to improve precision considerably.

Option 2 This option, when compared to the required run, evaluates the impact of swapping the

Templates									
Test	Possible	Actual	Correct	Incorrect	Spurious	Missing	Recall	Precision	Overgeneration
Required	108	143	62	0	81	46	57	43	57
Option 1	104	63	37	0	26	67	36	59	41
Option 2	105	125	44	0	81	61	42	35	65
Option 3	103	53	26	0	27	77	25	49	51
Option 4	107	108	56	0	52	51	52	52	48
Option 5	108	133	54	0	79	54	50	41	59

Table 3: Results from different tests for Template-id slot

Slot	Recall	Precision	Overgeneration
Template-Id	52	52	48
Incident-Type	50	95	0
Category	38	56	20
Org-Perps	25	16	76
Perp-Confidence	4	10	54
Phys-Target-Types	1	50	0
Human-Target-Types	7	43	4
Incident-Location	25	26	55
Phys-Effects	11	33	17
Human-Effects	4	28	28
Matched only	27	41	45
Matched/Missing	15	41	45
All Templates	15	23	69
Set Fills Only	18	57	20

Table 4: Detailed results for Option 4

Summary of Set Fills									
Test	Possible	Actual	Correct	Incorrect	Spurious	Missing	Recall	Precision	Overgeneration
Required	570	223	67	38	73	420	16	40	33
Option 1	547	61	35	3	9	495	8	69	15
Option 2	557	172	107	5	40	425	20	64	25
Option 3	544	48	37	0	8	504	7	77	17
Option 4	568	179	80	20	35	424	18	57	20
Option 5	571	208	139	4	41	404	25	68	20

Table 5: Results from different tests based on SET FILLS ONLY row

two training sets used for optimal_query versus the other concepts. The effects on recall and precision are insignificant.

Option 3 Option 3 used the same training sets as option 2 but the threshold was set to default. This led to a sharp drop in the total number of templates resulting in a much smaller value for recall. But as in option 1, there is significant improvement in precision. Comparing option 2 to option 3 serves the same purpose as comparing option 1 to the required run.

Option 4 (Official-3) This option used the Training Set 3 to develop the rule vectors for slot fills. When compared to option 2, this provides an assessment of the effectiveness of replacing Training Set 1 by Training Set 3 for learning rule vectors for various possible slot fills. This option results in a large number of templates compared to the other options. The use of Training Set 3 makes the examples in the training set relatively *cleaner*. The method retains the level of recall roughly at the same level as the Official-1 run and option 2 while improving precision.

Option 5 Option 5 is meant to test the effect of the use of phrases on the development of rule vector. The results from this test are similar to those from option 4. Since we have obtained other results (not reported) where the use of phrases is helpful, we feel that the results concerning phrases is not yet conclusive.

EXPLANATION OF TEST SETTINGS

In each experiment, two different training sets are employed. A new (test) message is processed against the optimal_query computed from the first training set. The message is processed with respect to the rule vectors corresponding to all the possible slot fills, as computed from the second training set. A message is deemed relevant to MUC-database either if it is sufficiently similar to the optimal_query or, based on the concepts that are applicable and the rules in the rulebase, the inference engine evaluates the root concept to be true. Since the second training set is used to develop rule vectors for different slot fills, it is desirable that the training set messages contain incidence of all the slot fills. If there are no examples corresponding to a slot fill, the system is unable to develop a rule vector for that fill and consequently, cannot recognize the occurrence of that slot fill in a new message. The way in which training sets are used enables us to test the effects of not only the size of the training set, but also the quality in terms of training messages being non-ambiguous and noise-free.

The second variable in the optional testing - the threshold activation value - is used to select or ignore a slot fill. The system compares the representation of each message with respect to the rule vectors and computes an activation value for the corresponding slot fill. The default activation value is taken to be 0, that is, a slot fill is deemed applicable if the activation value of its rule vector with respect to the message is positive. A precision-recall tradeoff can be achieved by changing the value of the threshold activation value. If this threshold is lowered, a concept becomes applicable to more messages resulting in an improvement in recall at the cost of precision.

EFFORT

The team for the MUC-3 project consisted of two professors, three graduate research assistants, and four part-time programmers. The following graduate students made significant contributions to this project: V. K. Elayavalli and Y. Zhang of USL, and S. K. Bhatia of UNL. The bulk of the effort was spent on the process for phrase extraction, followed by selection of training set, developing inference engine for the rulebase and the template filler. The learning and use of scoring program also took a considerable amount of time.

LIMITING FACTORS

The biggest limiting factor was time. Our estimation of the time and manpower for the project was also affected, in part, by a lack of participation in MUC-1 and MUC-2. The project often competed, usually unsuccessfully, for the time of graduate students because of their classes and examinations.

A lot of effort was spent on the extraction and the use of phrases. However, this effort did not provide much contribution as the phrase information was not exploited to its limits. Towards the end, we succeeded in developing interesting techniques for phrase extraction and usage but could not realize the benefits due to time constraints. In retrospect, we feel that we should have spent more time on template filler module than on indexing module.

TRAINING

The quality and size of training sets have tremendous effect on the performance of the system. A limitation of hardware affected the size of training set that could be selected. A large training set required larger main memory and computer time for different modules in the project than could be afforded by the limited computing resources at the two campuses. Again, the computing resources had to be shared with instructional and other research users which had an adverse effect on the resources for the project. Limited hardware resources were responsible for our search for a *good* training set. In this project, we were limited to use at most 300 messages in the training set due to memory and time constraints. Our initial approach was to manually select some messages to develop the training set. Later, we developed the training set through a program by selecting only those messages that addressed exactly one incident type.

Ideally, the training set should contain enough messages such that all possible set fills are sufficiently represented. If the training set does not contain any message addressing a certain slot fill, the system is incapable of recognizing that slot fill. We also developed a module which could be used to select a training set by computing the representational similarity of messages in the test set to those in the development set. Unfortunately, the module was not tested well enough to be used for the MUC-3 official testing.

DOMAIN INDEPENDENCE

Nearly all of the system can be used independent of the domain of application. The system automatically learns the rule vectors corresponding to different slot fills and uses these rule vectors to identify the slot fills in new messages. The only domain dependent part of the system is the rule base that is used to decide the relevance of a message depending on whether certain concepts are applicable to the message in a desired combination.

CONCLUSION

MUC-3 provided us with a unique opportunity to test our ideas on conceptual classification of documents in the area of message understanding. Our approach is based on the recognition of message contents rather than actually understanding the messages. The recognition of certain patterns in a message allows the system to conclude whether certain subjects are addressed in a message. The system is, however, highly sensitive to the selection of a good training set.

In the context of a MUC-like task, the system is capable of recognizing the presence or absence of different concepts that correspond to fills drawn from a set of values. Specifically, our system can efficiently identify the domain of a message and which among certain salient concepts are addressed. For example, in the case of Option 4 run, the recall and precision associated with the optimal_query vector is respectively 0.78 and 0.88 (i.e. if the question is whether at least one template should be generated). The performance, in terms of precision, is also impressive in certain slots such as

INCIDENT TYPE(S) and effects on PHYSICAL OR HUMAN TARGETS. Furthermore, the concept rule vectors are found to be successful in identifying relevant paragraphs within messages. In many application environments, such capabilities may be adequate. Furthermore, our system may be used as a front-end to a comprehensive message understanding system.

The system has only a limited ability to identify fills that are of string type appearing in the messages. Phrase extraction, combined with the locality of information, was particularly useful in filling certain slots that require string fills. The process to extract and use phrase information can be exploited to a greater extent than has been done in the present system.

The system can be improved by the following enhancements. From a domain independent viewpoint, the system can benefit from a more robust procedure for training set selection. Moreover, the process to extract and use phrase information can be exploited to a greater extent. For improving the performance in the current domain, the template filler module can be modified by taking into account the information regarding dependencies between different slot fills. In general, much more effort is needed in designing the template filler module.

PART III: SYSTEM DESCRIPTIONS

The papers in this section, which were prepared by each of the fifteen sites that completed the MUC-3 evaluation, describe the systems that were tested. The papers are intended not only to outline each system's architecture but also to provide the reader with an understanding of the effectiveness of the techniques that were used to handle the particular phenomena found in the MUC-3 corpus. To make the discussion of these techniques concrete, most of the sites make specific reference to some of the phenomena found in message TST1-MUC3-0099 from the dry-run test set and discuss their system's handling of those phenomena. The full text and answer key templates for that message are found in appendix H of the proceedings.

The sites were asked to include the following pieces of information in this paper:

* Background: how/for what the system was developed, and how much time was spent on the system *before* MUC-3

* Explanation of the modules of the system

* Explanation of flow of control (interleaved/sequential/...)

* Explanation (without system-specific jargon) of processing stages:
- Identification of relevant texts and paragraphs
- Lexical look-up (example of output and lexicon)
- Syntactic analysis (example of output and grammar)
- Semantic analysis (example of output and semantic rules)
- Reference resolution
- Template fill

* Sample filled-in template, with an explanation of interesting things:
- things system got right
- things system got wrong