PRC PAKTUS:
MUC-3 TEST RESULTS AND ANALYSIS

Cheryl Kariya
PRC Inc.
1500 Planning Research Drive
McLean, VA 22102
email: kariya_cheryl@po.gis.prc.com

For natural language understanding researchers at PRC, preparing for MUC-3 was a useful endeavor in many respects. The value of having a well-defined task, a large corpus, and an automated scoring program can hardly be overestimated. This exercise has pointed out the strengths of our system, confirmed our feelings about which aspects most need work, and taught us how to approach a task of this magnitude efficiently. In this paper, we will discuss our test results, the ways in which we prepared for the task, and lessons learned.

RESULTS

Table 1 summarizes PRC's scores for MUC-3 Phase 2.

SLOT	REC	PRE	OVG	FAL
template-id	88	61	39	
incident-date	51	60	0	
incident-type	69	78	0	1
category	68	58	24	30
indiv-perps	3	30	10	
org-perps	1	100	0	
perp-confidence	20	13	47	12
phys-target-ids	9	32	12	
phys-target-num	26	66	6	
phys-target-types	10	67	0	0
human-target-ids	2	42	0	
human-target-num	4	100	0	
human-target-types	2	70	0	0
target-nationality	6	100	0	0
instrument-types	0	*	*	0
incident-location	47	53	0	
phys-effects	18	24	41	2
human-effects	5	21	14	1
----------	---	----	----	----
MATCHED ONLY	32	53	20	
MATCHED/MISSING	28	53	20	
ALL TEMPLATES	28	36	46	
SET FILLS ONLY	28	47	24	1

Table 1: PRC Scores

Overall, we were pleased with PAKTUS's capabilities and performance, and we are excited about planned improvements to the system. It was particularly satisfying to observe that

only a few changes had to be made to the grammar, and that the lexicon structure and tools easily accomodated new additions.

It came as no surprise to us that the discourse module was the component most in need of development, as it is the most recent and least-developed part of the system. Pieces for a fairly major restructuring are already in place, and discourse development will be one of our main efforts in the coming year. To a large extent, our overgeneration of templates was due to the unfinished state of the discourse component. Currently, PAKTUS proposes one template for each sentence which contains an interesting-looking incident. In discourse processing, the incident templates are merged for the same incident type, if the date and location are not missing and not different. This was a temporary solution designed to prevent wild overgeneration. As we achieve greater recall in the other slots, those fills can be used in a more sophisticated merging strategy; for the moment, a simple strategy was all that was feasible.

The main limiting factor for PRC was the availability of people for development. CPU cycles was also a factor, insofar as it limited the number of development tests that could be run in a given amount of time. It took from 4 to 6 hours to process 100 documents, depending on trace options.

We directed most of our energies to linguistic development, as shown in Table 2 below, and except for the discourse component, the linguistic aspects of the task have essentially been completed. Because we had less time remaining to devote to engineering issues, much of the information that PAKTUS produced has not yet been used in template-filling. In using the linguistic output to fill templates, we began with the two aspects of the template-filling task we considered the most important: identification of relevant incidents (template id) and identification of incident type. Our recall and precision scores for those slots reflect the amount of time we spent on them (see Table 1). We believe that, given more time to convert linguistic output to template fills, we will be able to achieve comparable scores for the other slots as well.

TIME SPENT ON DEVELOPMENT

Five PRC researchers participated in MUC-3 development. Table 2 shows an estimate of our level of effort for MUC-3 and a breakdown of tasks.

	HOURS	% of TOTAL	% of LING DEVEL
LING. DEVEL.			
preprocessor	284	11	22
grammar	154	6	12
lexicon	457	18	35
discourse	100	4	8
linguistic troubleshooting	300	11	23
TOTAL	1295	50	100
ENGINEERING & TECHNICAL TROUBLESHOOTING			
system integration	150	6	
back-end engineering	300	11	
technical troubleshooting	308	12	
TOTAL	758	29	

	HOURS	% of TOTAL	% of LING DEVEL
MISC.			
technical direction	80	3	
other*	467	18	
TOTAL	547	21	

*learning emacs, scoring program, committee work, papers, presentations

Table 2: Breakdown of MUC-related hours

PAKTUS DEVELOPMENT

In adapting PAKTUS (PRC's Adaptive Knowledge-based Text Understanding System) to any new domain, adjustments to each of the core system components must be made. We briefly discuss the types and amounts of adjustments that were necessary in each component, and outline some of the new features that were added as we worked on MUC-3.

Preprocessor

As for any new message type, a new template specifying the format of the input stream had to be built, so that the input could be separated into messages. Methods for handling corpus-specific use of punctuation -- for example, [square] brackets and double dashes -- were developed. Header information (time and location) was saved for later processing. Also, because time (date of incident) played such an important role in the MUC-3 task, a new feature was added to the preprocessor which bracketed time expressions in the text, calculated a date, and passed it on to the parser as a symbol. In general, adaptations to the preprocessor are only a small proportion of the entire development effort.

Lexicon

Adding the domain-specific lexicon required a considerable amount of time. This is primarily due to the volume of new items to be entered. Especially time-consuming was identifying the terrorist organizations (due to variation in names, acronyms, translations) and their associated countries. PAKTUS's automated tools for entering words, synonyms and compounds, together with feature inheritance in the lexical and semantic networks, made actually entering individual items easy. Because the MUC-3 corpus contains large numbers of similar items (e.g., locations, terrorist organizations), facilities for batch-entering words were developed. In addition to adding lexical items, a few features had to be added to categories of words; for example, a slot was added to words in the PERSON category to aid in classifying the TYPE of human target. Again, inheritance in the network made this process straightforward and fast.

Another new feature for the lexicon is the use of heuristics for guessing at unknown words whose roots are unknown. (PAKTUS uses inflectional and derivational morphology if the roots are in the lexicon.) Word regularities are identified using forward and reverse concordances of the MUC-3 corpus, and exceptions to those regularities are entered in the lexicon. An example of a heuristic used for MUC-3 is the guess that an unknown word ending in "-z" or "-o" is a Spanish name. We have thus far developed about two dozen heuristics.

Grammar

The only significant change made to the core grammar for MUC-3 involved time expressions. A new arc was added to handle bracketed time expressions as adverbs, and the case-frame-applier was also adjusted. Some additional work was done on apposition, but this was not corpus-specific development. Altogether the grammar was well-suited to handle a MUC-3-like task, and changes were minimal.

Discourse

PAKTUS's discourse component is currently fairly application-specific. Many of the patterns which identify important information (for this task) had to be written from scratch, although a few were recycled from previous applications, such as MUC-2. The discourse component, both for MUC-3 and in general, is, as we said above, the least-developed aspect of PAKTUS, and the area in which we expect to make the most improvement in the coming year. One new feature already implemented is the addition of "word patterns," which use only lexical information (but include conceptual associations), to supplement the discourse patterns, which use parse output. These word patterns accounted for roughly one-third of our recall.

Future directions

We put some effort into developing a pattern-based filter, to be used at the pre-parse stage, for identifying relevant sentences and/or documents. This filter was not used in the tests, but we expect to complete development and implement it for MUC-4. In addition, we plan to expand the bracketing capabilities to include at least names, titles, and locations. The major thrusts in the coming year, however, will be:
- to modify the discourse component to include more broadly-based, linguistic information; and
- to develop more tools for analyzing our output.

Reusability

One of the best aspects of the MUC-3 corpus is its generality. Nearly all of the development we did for this effort can be reused for another application. The exceptions would be some domain-specific lexicon and some of the specific discourse patterns.

TRAINING AND IMPROVEMENT

We relied almost exclusively on the 100 messages in Test 1 for training after the February interim conference, as it had a reliable, consistent key. Using Test 1, we ran approximately 30 tests. A few of these were run to determine the effect of different timing strategies on output; the others, to test improvements in slot-filling from using lexical patterns, partial parses, run-ons, etc. The development corpus was used primarily for lexical development. Two large tests were run on the entire development corpus to locate a few coding errors and measure linguistic performance.

The improvement in PAKTUS's linguistic performance between February and May 1991 can be seen in the following tables, derived from the test runs on the development corpus.

February results

	total	completed parses	partial parses	run-ons	failed
Total # of sentences	18016	7039	5460	4887	630
% of sentences	100	39.1	30.3	27.1	3.5
Total time (%)	100	12.7	41.6	45.7	N/A
Avg time for parse (in seconds)	7.0	2.3	9.5	11.7	N/A
Avg time for preprocessor (in seconds)	0.4	0.3	0.4	0.5	N/A

May results

	total	completed parses	partial parses	run-ons	failed
Total # of sentences	18584	8741	4263	5074	506
% of sentences	100	47.0	22.9	27.3	2.7
Total time (%)	100	17.7	34.5	47.9	N/A
Avg time for parse (in seconds)	6.3	2.4	9.5	11.1	N/A
Avg time for preprocessor (in seconds)	2.5	2.1	2.9	3.0	N/A

The most significant changes is that the number and percentage of sentences that were fully parsed went up nearly 8% between February and May. This reflects changes in all PAKTUS components except the discourse components. Comparison between PRC's scores for the Test 1 and Test 2 reflects the discourse improvements as well.

WHAT WE LEARNED

About PAKTUS

As PAKTUS had never before been exercised on a scale as large as the MUC-3 corpus, this was an opportunity to find out how it would hold up in a life-sized scenario. We learned that the system architecture, knowledge representation and algorithms were more than adequate for the task, that it was possible to do all we needed to do, with no major changes. Further, we discovered just how robust PAKTUS is. We were pleasantly surprised, for example, that PAKTUS was able to parse 47% of the sentences in the development corpus, particularly since relatively little time was spent on grammar development for MUC-3.

The MUC-3 corpus and automatic scoring tool made it possible for us to do extensive experimentation on PAKTUS's timer. We discovered several points in the parsing where speed could be improved, while losing little or no important information.

Although we had been working on extensions to the preprocessor (e.g., document/sentence filtering, word patterns) before the MUC-3 Phase 1 conference, participation in MUC-3 helped crystallize our ideas on how best to use those extensions. Further work showed us that preprocessor output could be used in many profitable ways (e.g., helping to resolve anaphora), without compromising linguistic principles.

Finally, as mentioned above, our participation in MUC-3 gave added impetus to the development of an improved discourse component.

About the task

Working on MUC-3 provided the PRC team of researchers valuable experience in large-scale system development. Basically, we learned what needed to be done and how to divide the work among ourselves efficiently. We also, albeit somewhat belatedly, learned to make maximum use of the tools provided us. For example, rather late in the game, it surprised us that we had not thought to use the development corpus as a source of information about perpetrator organizations!

About evaluation

Task design: Complete specification of a complex task to be performed is a non-trivial undertaking, requiring multiple iterations to resolve outstanding issues. MUC-3 has clearly demonstrated that, in spite of excellent initial task design, unforeseen issues inevitably arise, and an ongoing mechanism to provide clarification is indispensable for such complex problems.

Training: A valid evaluation must be representative of the domain and type of text on which the system was trained. The training corpus must therefore be sufficiently large and varied to cover most of the issues which are likely to arise in a test. (This is one of many areas where MUC-3 was vastly superior to MUC-2.) Furthermore, for complex tasks such as MUC-3, it is critical to have an authority available to judge the correctness of a system's response to the training corpus. The development corpus keys, to which each site contributed 100 messages' worth, provided such an "authority". The process of manually generating keys pointed out many issues which required task clarification, and thus reduced the risk of misinterpreting some aspects of the task.

Test Set: Selection of test messages should be done, as it was in MUC-3, from the corpus before development begins. How to ensure that the test set is representative of the corpus, linguistically and in terms of content, is a question which it may be useful to address for MUC-4, as is the issue of the appropriate size of a test set. The "answers" to the test set should be produced independently of the test, by an unbiased party (not the system developers). Agreement by all participants to abide by the decisions of the unbiased party must be obtained before the test, and must be adhered to after the test.

Metrics and Scoring: The automatic scoring program is a valuable tool for both development and testing. It ensures a base level of impartiality in scoring, although that impartiality is mitigated by the fact that participants could assign themselves credit interactively, possibly using different standards. A possible solution, ultimately adopted for MUC-3, is to have all results scored blindly by an unbiased party.

The MUC-3 scoring program has been useful not only as a means for standardization in testing, but also as a development tool. The fact that different metrics (e,g., recall, precision, overgeneration) were provided on a slot-by-slot basis gave us a clear picture of our progress. For viewing test results, this means of reporting highlights the strengths and weaknesses of different aspects of the systems, rather than providing a single numeric score which masks the details. It thus provides a larger, fairer picture of the systems than would otherwise be possible.