NEW YORK UNIVERSITY PROTEUS SYSTEM: MUC-3 TEST RESULTS AND ANALYSIS

Ralph Grishman, John Sterling, and Catherine Macleod

The PROTEUS Project
Computer Science Department
New York University
715 Broadway, 7th Floor
New York, NY 10003

{grishman,sterling,macleod}@cs.nyu.edu

RESULTS

Our "max-tradeoff" run achieved a recall of 44% at a precision of 57%:

	REC	PRE	OVG	FAL
MATCHED ONLY	49	57	20	
MATCHED/MISSING	44	57	20	
ALL TEMPLATES	44	36	50	
SET FILLS ONLY	40	62	18	1

Although the template specifications call for us not to produce templates for non-specific event descriptions, our base run did not attempt to filter out such descriptions, in part because the criteria are hard to specify, but mainly because at our low recall there was a fair chance of missing date or perpetrator information and thus eliminating templates incorrectly. However, we made two optional runs to attempt to eliminate these templates. For both runs, we eliminated all military and terrorist targets (we had found from our training runs that these targets were frequently being incorrectly combined with civilian targets and therefore emitted as parts of templates). For the first run, we then eliminated all templates with neither perpetrator nor targets (typically arising from nominalizations such as "the recent attack", when there was no preceding attack being referred to); the results were:

	REC	PRE	OVG	FAL
MATCHED ONLY	50	60	17	
MATCHED/MISSING	43	60	17	
ALL TEMPLATES	43	43	41	
SET FILLS ONLY	39	62	18	0

For the second run, we eliminated all templates with no targets; this produced:

	REC	PRE	OVG	FAL
MATCHED ONLY	50	60	17	
MATCHED/MISSING	40	60	17	
ALL TEMPLATES	40	45	37	
SET FILLS ONLY	37	61	18	0

Since this is a process of template elimination, the changes are most noticeable on the "all templates" line, where the precision rises from 36% to 45%. The effect can also be seen in template overgeneration (not shown in the above tables), which falls from 48% on the max-tradeoff run to 30% on the final run.

HOW OUR TIME WAS SPENT

Our time prior to the interim meeting was basically spent porting the system we had previously developed for MUC-2 to the new domain. This involved extending the grammar, developing a new concept hierarchy and lexico-semantic models, and writing an initial template generator. Our system was marginally operational at the time of the interim meeting; our two "official" runs were at 14% and 21% recall. We recognized that major enhancements were needed in at least three areas: handling complex sentences; enriching the domain model; and extending the template generator (which did not handle all the slots, much less all the cases for each slot, at the interim evaluation). These occupied most of our time in March and April.

HANDLING COMPLEX SENTENCES

The sentences of MUC-3 were consistently more complex than those from the military messages we processed for MUC-1, MUC-2, and other tasks. We took several measures in the syntactic analyzer to accomodate this increased complexity:

- We allowed for the skipping of words through a very small extension to our grammar. Basically, we added productions to allow three symbols -- sentence adjunct, right adjunct of noun, and pre-nominal adjective -- to match any sequence of input tokens. Using the parser's scoring mechanism, we imposed a substantial penalty for each token so matched (with the exception that a group of tokens enclosed in parentheses, brackets, or dashes were penalized as a single token). In this way, if no full sentence parse was possible, we would search for a parse skipping the minimal number of sentence words.
- In the event that no parse of the entire sentence could be obtained, our MUC-2 system had a fall-back which sought the longest substring starting at the first word for which a parse could be produced. We extended this to reanalyze the *remainder* of the sentence and find the longest sentence or noun phrase in that substring.
- Our earlier system propagated all ambiguities up the tree: if there are two analyses of words 10 to 15 as a noun phrase, and this noun phrase can be incorporated into a clause, we will generate two separate clause analyses. In our current system we "factor" these ambiguities, taking only the highest scoring analysis of a node spanning a particular string of tokens. This sometimes produces worse analyses when an ambiguity cannot be resolved locally but appears to more than make up in terms of speeding the analysis of long sentences.

This combination of techniques was quite effective in getting through the corpus. Our lexical scanner identified 1561 sentences in the TST2 corpus. Of these, only 609 contained keywords which led our system to attempt a full syntactic and semantic analysis of the sentence (our system attempts a full analysis only if a sentence contains a word which may be relevant to the template-filling task, such as a type of attack or damage or the name of a terrorist group). Of the 609 sentences attempted, 511 (84%) obtained a full sentence analysis.[1] Of the remaining 98 sentences, the system was able to analyze at least an initial substring for 89.

We also relaxed the operation of the semantic analyzer in several regards. The MUC-2 analyzer was intentionally quite conservative: if a high-level structure in the parse tree did not have a semantic model, we did not attempt to analyze the semantics of embedded structures. Furthermore, if the high-level structure did have a model, but certain arguments or modifiers in the tree were not matched in the model, those arguments were ignored. This approach made sense in a context where we could expect to have semantic models of most of the constructs pertinent to the template-filling task. Since that was not the case this time, at least in our six-month time frame, we took a more inclusive approach. Lower-level structures were analyzed even if the higher-level structure had no semantic model; operands and modifiers not matched by the model were analyzed semantically as isolated entities. In addition, we relaxed the criterion for matching a model: if a required argument was present but with the wrong semantic class, we allowed the match with a penalty (so that a predicate would be generated and the other arguments processed) but did not incorporate that argument into the predicate structure.

THE TRAINING PROCESS

Figure 1, which plots our total recall score (matched/missing) over time for several document sets, shows a gradual, steady improvement in our system's performance. The graph is somewhat ragged in February and early

[1] These 511 included sentences where one or more words were skipped. Because of the penalty structure, however, it was rare that -- except for sequences enclosed in parentheses, brackets, or dashes -- more than two words would be skipped in a single sentence.

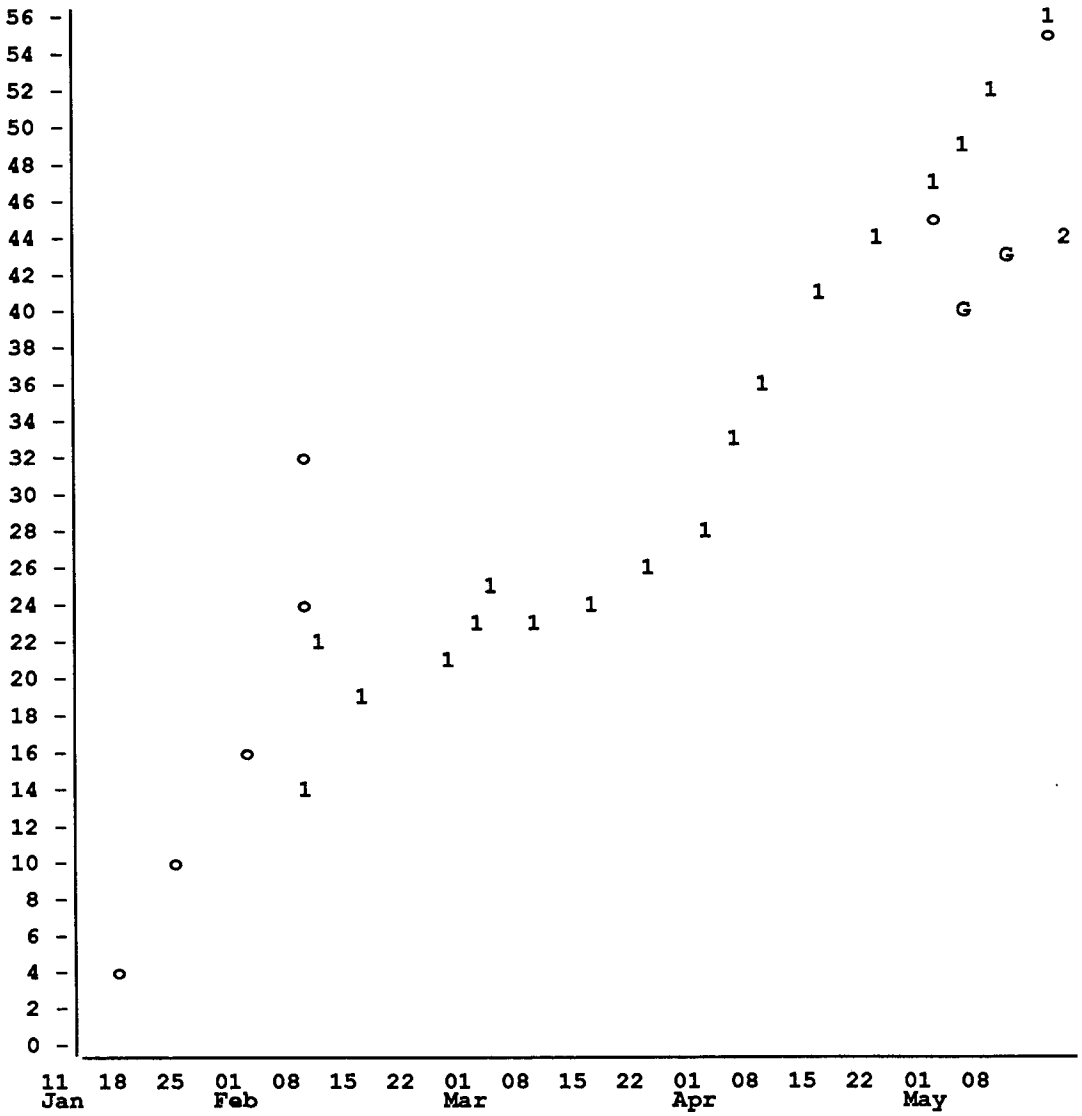

Figure 1. System total recall score (matched/missing) as a function of time during MUC-3 development. Different symbols represent different test corpora: o = DEV-0001-0100; G = DEV-0601-0700; 1 = TEST1; 2 = TEST2.

March while we experimented with different strategies for handling complex sentences (as described above). After these strategies were in place, we resumed our steady climb of several percent per week. This reflects a gradual growth in the concept hierarchy and lexico-semantic models, gradual enhancement of the template generator, and to a lesser extent additions to the grammar and semantic analysis procedures (e.g., handling non-restrictive modifiers for reference resolution). Surprisingly, perhaps, the curve has not yet started to flatten, which suggests that the performance of our system (and probably most systems) was still largely time limited. As TEST2 pointed out (but we already fully realized), there were still substantial gaps in our lexico-semantic models and (to a lesser degree) in our template generation procedure.

The development of the concept hierarchy and lexico-semantic models was based on a review of the usage of terms throughout the entire corpus (effectively using keyword-in-context indexes). The development and debugging of the system as a whole was based on a more intensive study of 200 documents, the TST1 corpus and the

NOSC DEV corpus (messages DEV-0001-0100). We ran and scored at least one of these two sets nearly every night to monitor system performance and detect bugs which crept in. In addition, we used a third set (DEV-0601-0700), labeled "G" on the graph, as our blind test (texts we had not studied except for keyword-in-context searches), and it was a fairly good predictor of our eventual TEST2 performance.

LESSONS

Portability

Our overall system structure was not substantially changed from the earlier system we had used to process military messages. Many of the enhancements we made would be applicable to other tasks involving general English input: use of a commercial machine-readable English dictionary, broadening the grammar, extending the semantics to handle a wider range of lingustic constructs, and adding various mechanisms for complex constructs, as described above.

Because of the specialized nature of our prior domains, our concept hierarchies and lexico-semantic models were essentially redone from scratch each time. If we are to address more tasks involving a broad range of everyday linguistic and semantic constructs, it will make sense to have in our system a core of lexical semantic knowledge, covering such things as copulas, performative and reporting verbs, causality, dates, times, and locations.

Parsing

The best-performing systems at MUC-3 all employed some form of syntactic and semantic analysis, but there was a wide variation in the type of syntactic analysis used, from full-sentence analysis (such as our system) to much more local syntactic analysis. It appears that -- as long as adequate recovery mechanisms were incorporated in the analyzer -- this variation did not have much effect on overall system performance. This is perhaps not surprising since most of the information crucial for template filling can be obtained locally. In a sentence such as `THE NEWSPAPER STATED THAT THE ALLEGED RIFT BETWEEN THE MILITARY OFFICERS BEGAN ONCE IT WAS DISCOVERED WHO WAS RESPONSIBLE FOR THE DEATH OF THE SIX JESUIT PRIESTS, IN WHICH ONE COLONEL, TWO LIEUTENANTS, AND SIX SOLDIERS WERE CHARGED.` the information before `THE DEATH OF ...` (the top few levels of the parse tree) adds little to the template-filling task. While full-sentence parsing may provide a more systematic solution for those cases where higher-level information is needed (e.g., sentences such as "X charged that ...", where the confidence of the information is affected), the effect on overall performance may be evident only once we have substantially more complete semantic models.

Improvements

There are certainly aspects of the system which were fixed to the point where they were adequate for the MUC-3 evaluation but cannot be regarded as satisfactory for the long term. The multiple mechanisms for coping with complex sentences did cope, in most cases, but a simpler mechanism -- based perhaps on different search order and statistically-weighted grammar -- would be desirable. The concept hierarchy is quite primitive and should be enriched. The system maintains only the crudest of models of the overall discourse structure beyond a single sentence, essentially keeping track of the most recently mentioned attack. Though this doesn't seem to be a major limiting factor at present (compared to the gaps in our lexico-semantic models, for example), we expect that it will be more important in the future.

Perhaps most crucial, though, are better tools for acquiring lexico-semantic models. In particular, we might hope to have tools based on parsed text in place of our currently methodology based on the raw text. We have done and are continuing to do experiments with tools for semantic class and semantic pattern acquisition, and we expect that these tools will play a role in helping us to prepare a richer set of semantic models for MUC-4.