The LINK System: MUC-4 Test Results and Analysis

Steven L. Lytinen, Sayan Bhattacharyya, Robert R. Burridge, Peter M. Hastings, Christian Huyck, Karen A. Lipinsky, Eric S. McDaniel, and *Karenann K. Terrell*
Artificial Intelligence Laboratory
The University of Michigan
Ann Arbor, MI 48109
E-mail: lytinen@caen.engin.umich.edu

Results

The University of Michigan's natural language processing system, called LINK, was used in the Fourth Message Understanding System Evaluation (MUC-4). LINK's performance on MUC-4's two test corpora is summarized in figure 1.

Although we only tested LINK in a single configuration, there were several parameters that could have been varied in the system. They include the following:

1. **What to do with undefined words.** When the system identified a group of undefined words as a likely noun phrase, it was assumed that this noun phrase referred to some kind of HUMAN or PLACE.[1] Thus, these noun phrases were potential candidates to fill the LOCATION, PERP, PHYS TGT, or HUM TGT fields of a template.

2. **When to generate templates.** A template was only generated if an appropriate filler for the PERP, PHYS TGT or HUM TGT field had been extracted from the text.

3. **When to merge templates.** Every time a new template was generated for an article, the system considered merging it with existing templates. A merge was performed if another template with the same INCIDENT TYPE already existed, and if there were no explicit contradictions between the existing template's filled fields and the new template. For example, if the two templates had different DATE fields, they were not merged. In addition, BOMBING and ATTACK templates were merged if they had no contradictory fields.

Amount of effort

We estimate that 1.5 person-years were spent on our MUC-4 effort. Figure 2 shows the breakdown of this effort on different parts of the system.

Prior to MUC-4, LINK had been used in several smaller-scale applications, including the extraction of information from free-form textual descriptions of automobile malfunctions and the repairs that were made to fix them; as well as an application involving free-form textual instructions for assembly line workers.

Little modification was required of the parser itself for MUC-4. However, several new modules were built around the parser. In particular, since both of our prior applications involved

[1]See our accompanying system summary paper for details.

```
TST3     SLOT                         POS  ACT|COR PAR INC|ICR IPA|SPU MIS  NON|REC PRE OVG FAL
         -----------------------------------+-----------+-------+------------+----------------
         MATCHED/MISSING             1540 1103|557 155 141|  6 101|250 687 1071| 41  58  23
         MATCHED/SPURIOUS            1117 1588|557 155 141|  6 101|735 264 1135| 57  40  46
         MATCHED ONLY                1117 1103|557 155 141|  6 101|250 264  668| 57  58  23
         ALL TEMPLATES               1540 1588|557 155 141|  6 101|735 687 1538| 41  40  46
         SET FILLS ONLY               741  549|303  58  63|  0  36|125 317  488| 45  60  23   1
         STRING FILLS ONLY            398  249|118  20  40|  4  20| 71 220  299| 32  51  28
         -----------------------------------+-----------+-------+------------+----------------
                                           P&R        2P&R      P&2R
         F-MEASURES                        40.49      40.2      40.8

TST4     SLOT                         POS  ACT|COR PAR INC|ICR IPA|SPU MIS  NON|REC PRE OVG FAL
         -----------------------------------+-----------+-------+------------+----------------
         MATCHED/MISSING             1188  730|374 121  99|  8  63|136 594  802| 36  60  19
         MATCHED/SPURIOUS             764 1264|374 121  99|  8  63|670 170  976| 57  34  53
         MATCHED ONLY                 764  730|374 121  99|  8  63|136 170  473| 57  60  19
         ALL TEMPLATES               1188 1264|374 121  99|  8  63|670 594 1305| 36  34  53
         SET FILLS ONLY               580  357|211  35  49|  3  14| 62 285  362| 39  64  17   0
         STRING FILLS ONLY            307  171| 88  19  25|  1  18| 39 175  227| 32  57  23
         -----------------------------------+-----------+-------+------------+----------------
                                           P&R        2P&R      P&2R
         F-MEASURES                        34.97      34.38     35.58
```

Figure 1: LINK's performance on the TST3 and TST4 corpora

reading only single-sentence texts, with no need to monitor context, there was a need to enhance the system so that multi-sentence texts could be processed. The reader is referred to our accompanying system summary paper for a description of each module.

Development time was definitely the limiting factor in our system's performance. Although we felt that our knowledge base was approaching completion toward the end of the development time, considerably more effort could have been expended toward improving our system's ability to handle multi-sentence input had more time been available. We will discuss this further in section .

Tokenizer	2
Preprocessor	2
Knowledge base development	9
Postprocessor:	
Template generation	3
Template merging	1
Reference resolution	1

Figure 2: Breakdown of MUC-4 effort by module (person-months)

Training of the system

We used the MUC-3 development corpus answer keys to help develop the knowledge base for our system. Some of this development was partially automated, although not as much as we had originally hoped. The answer keys contained a great deal of information about how various lexical items should map to the HUM TGT, PHYS TGT, and INSTRUMENT TYPE fields in the MUC-4 templates. For example, the appearance of LAW ENFORCEMENT: "POLICEMEN" in field 20 of several answer key templates, along with PLURAL: "POLICEMEN" in field 21, suggested that "POLICEMEN" should be defined in our lexicon as a plural noun which means LAW ENFORCEMENT. We were able to use this information to define a substantial percentage of the nouns in our lexicon.

Unfortunately, there was no such source of information for other types of words that were of interest in the domain, such as verbs, adjectives, prepositions, etc. An INCIDENT: DESCRIPTION field in the template would have provided information for verbs, but no field existed in the MUC-4 templates. Thus, the remainder of the lexicon was constructed entirely by hand. Our test configuration system contained a total of 6700 lexical entries, with 7532 distinct definitions (i.e., some words were defined with more than one sense).

The system's grammar was also developed by hand. The grammar in the test configuration of our system contained 565 rules. Although many rules were not related to the terrorism domain, and thus could presumably be used in a different domain, about half of the rules were domain-specific, and could not transfer to a new domain without some inspection and modification. For example, rules about combining noun groups often contained semantic information which was specific to the domain (e.g., a noun meaning BOMB followed by a noun meaning ATTACK maps to a BOMBING with the INSTRUMENT field filled by the BOMB noun).

What worked

In a large-scale natural language application such as MUC-4, it is virtually certain that an NLP system will not be able to produce a complete syntactic and semantic analysis for multi-sentence or multi-paragraph articles. Developing a complete lexicon, grammar, or set of semantic interpretation rules for such an application is virtually impossible. Thus, it is very important for a system to have strategies to deal with texts which cannot be completely processed. Our system's strategies for incomplete processing were vital to its ability to perform at the level that it did. These strategies included the following:

1. **Preprocessing: identifying noun phrases.** The preprocessor, explained in detail in our system summary paper, grouped together words which were candidate noun phrases. These NP's often included words which were not in the system's lexicon. As a result, undefined words did not interfere with the system's ability to parse a sentence. Although our lexicon contained 6700 entries, we estimate that nearly 14,000 distinct lexical items appear in the MUC-3 training corpus. Thus, an effective approach for dealing with undefined words was critical to our system's performance.

2. **Identifying important partial parses.** Even with the enhancement provided by the preprocessor, our system did not succeed in parsing the majority of sentences that it encountered. However, information was extracted from these sentences by examining the

constituents that were built, even though they did not lead to a complete parse. This ability was vital to the performance of our system, and is described in more detail in our system summary paper.

What didn't work

Our system's ability to correctly integrate information extracted from multiple sentences was its weakest point. Most of the decisions as to how information should be integrated were made in the postprocessor; thus, this module is clearly the best candidate for rewriting.

Several problems existed in the postprocessor. First, its strategies for deciding when two templates should be merged were not very effective. As described earlier, this decision relied purely on the information contained in the two templates which were being considered for merging. By default, templates were merged unless the information they contained explicitly contradicted each other. This resulted in templates being merged even when the text contained obvious cues that two separate events were being described. For example, if a BOMBING template had already been generated for an article, a sentence beginning with "Another bombing occurred ..." would not generate a second bombing template unless information about LOCATION, PHYS TGT, etc., contradicted information in the first template.

Related to our system's poor merging heuristics was its lack of a sophisticated reference resolution strategy. Two kinds of reference resolution existed in the system, for names and pronouns. Whenever a name of a person was identified in the text, a list was searched for previous occurrences of that name, or of a longer name containing the new name. If a match was found, additional information about the person, which could be used to fill the DESCRIPTION or TYPE field, could be obtained from the prior mention of that person.

Pronominal reference in our system was extremely simplistic. When a pronoun was encountered, its referent was resolved to the most recent NP prior to it in the text which met simple semantic restrictions. If the pronoun was assigned to be the PERP of an event, then its referent had to be a type of TERRORIST. If it was assigned to be the HUM TGT, then its referent had to be a HUMAN who was not a TERRORIST. These simple heuristics obviously could have been improved greatly.

Finally, additional information about a template which appeared in a subsequent sentence often was not extracted. Lists of victims, additional information about perpetrators or victims, and so on that appeared in a separate sentence from the initial mention of a terrorist act were not usually added to the template.

What we learned

Perhaps the most important lesson of MUC is that in a large-scale natural language application, it is not yet possible to construct a knowledge base which will enable complete processing of even a majority of input texts. The domain is simply too large, and the possible variations in language too great. Thus, as we said earlier, it is very important for a system to have robust strategies for dealing with texts which cannot be completely processed.

Due to time constraints, we devoted very little effort to discourse processing. The lesson we learned here was twofold: on the one hand, we were a bit surprised that we could achieve even 40% recall with only the simplest heuristics for integrating information from multiple sentences.

Single sentences often contained enough information for our system to generate a template with sufficient information to match the answer key. On the other hand, we felt that we were nearing the maximum score that we could have achieved without further developing this aspect of our system. Thus, in another MUC-like task our group would devote a great deal of our effort in this area.

Finally, as we analyzed our system's results during development, we realized that the recall and precision scores used for evaluation would change significantly with relatively minor adjustments in the criteria used by the scoring program. Perhaps the prime factor that affected our own score was the criteria for what constituted a match between a response template and the answer key. Our system often erroneously merged two templates into a single template. Thus, correct fills of PHYS TGT and HUM TGT fields were often split between two templates in the answer key. At other times, our system generated two or more templates when a single template should have been generated. In this case, although correct information was split between the response templates, the scoring program only allowed a single match between response templates and the answer key, and counted additional response templates as spurious, even though they might have contained information which matched some of the information in the single template in the answer key.