# BBN PLUM:
# MUC-3 Test Results and Analysis

*Ralph Weischedel, Damaris Ayuso, Sean Boisen,*
*Robert Ingria, Jeff Palmucci*

BBN Systems and Technologies
10 Moulton St.
Cambridge, MA 02138
weischedel@bbn.com

## INTRODUCTION

Perhaps the most important facts about our participation in MUC-3 reflect our starting point and goals. In March, 1990, we initiated a pilot study on the feasibility and impact of applying statistical algorithms in natural language processing. The experiments were concluded in March, 1991 and lead us to believe that statistical approaches can effectively improve knowledge-based approaches [Weischedel, et al., 1991a, Weischedel, Meteer, and Schwartz, 1991]. Due to nature of that effort, we had focussed on many well-defined algorithm experiments. We did not have a complete message processing system; nor was the pilot study designed to create an application system.

For the Phase I evaluation, we supplied a module to New York University. At the time of the Phase I Workshop (12-14 February 1991) we decided to participate in MUC with our own entry. The Phase I Workshop provided invaluable insight into what other sites were finding successful in this particular application. On 25 February, we started an intense effort not just to be evaluated on the FBIS articles, but also to create essential components (e.g., discourse component and template generator) and to integrate all components into a complete message processing system.

Although the timing of the Phase II test (6-12 May) was hardly ideal for evaluating our site's capabilities, it was ideally timed to serve as a benchmark prior to starting a four year plan for research and development in message understanding. Because of this, we were determined to try alternatives that we believed would be different than those employed by other groups, wherever time permitted. These are covered in the next section.

Our results were quite positive, given these circumstances. Our max-tradeoff version achieved 45% recall and 52% precision with 22% overgenerating (See Figure 2.) PLUM can be run in several modes, trading off recall versus precision and overgeneration. Our other official run shows this tradeoff when PLUM is more conservative in generating templates. In this mode, we achieved 42% recall, 58% precision, and only 14% overgeneration. (See Figure 3.) This conservative version is actually our preferred mode of running the system. By being more conservative, recall dropped only 3 points, while precision increased 7 points and overgeneration was cut by one third.

## KEY SYSTEM FEATURES

Two design features stand out in our minds: fragment processing and statistical language modelling. By *fragment processing* we mean that the parser and grammar are designed to find analyses for a non-overlapping sequence of fragments. When cases of permanent, predictable ambiguity arise, such as a prepositional phrase that can be attached in multiple ways or most conjoined phrases, the parser finishes the analysis of the current fragment, and begins the analysis of a new fragment. Therefore, the entities mentioned and some relations between them are found in every sentence, whether syntactically ill-formed, complex, novel, or straightforward. Furthermore, this parsing is done using essentially domain-independent syntactic information.

The second key feature is the use of statistical algorithms to guide processing. Determining the part of speech of highly ambiguous words is done by well-known Markov modelling techniques. To improve the recognition of Latin American names, we employed a statistically derived five-gram (five letter) model of words of Spanish origin and a similar five-gram model of English words. This model was integrated into the part-of-speech tagger.

Another usage of statistical algorithms was an statistical induction algorithm to learn case frames for verbs from examples.

Major system components are shown in a diagram in the system handout. A more detailed description of the system components, their individual outputs, and their knowledge bases is presented in a companion paper [Weischedel, et al., 1991b]. We expect the particular implementations to change and improve substantially during the next two years of research and development.

After the message header has been processed, each sentence of the text is processed by linguistic components. Morphological processing includes a probabilistic algorithm for labelling both known and unknown words by part of speech. The most likely alternatives are passed to the MIT Fast Parser (MITFP), a deterministic parser designed to quickly produce analyses of non-overlapping fragments if no complete syntactic analysis can be found [deMarcken, 1990]. [1]

The semantic interpreter finds a semantic analysis for the fragments produced by MITFP. Semantic analysis is shallow in that some analysis must be produced for each fragment even thought most of the words in an article have no representation in the domain model. (For instance, Jacobs et al. [1991] estimates that 75% of the words in these texts are not relevant.) The semantic interpreter uses structural rules, almost all created after 25 February 1991. Nearly all of these carry over to all new domains. Domain-dependent, lexical semantic rules contain traditional case frame information. The novel aspect here is that the case frames for verbs were hypothesized by a statistical induction algorithm [Weischedel, et al., 1991a]. Each hypothesized case frame was personally reviewed over a two day period.

The discourse component performs three tasks: hypothesizing relevant events from the diverse descriptions, recognizing co-reference, and hypothesizing values for components of an event. The challenges faced by the discourse component are that syntactic relations present in the text and signifying the role of an entity in a hypothesized event are often not found by MITFP, and that reference resolution must be performed with limited semantic understanding. Given these challenges, it is clear from the test results that the discourse component does reconstruct event structure well, in spite of missing syntactic and semantic relations.

The template generator has three tasks: finding and/or merging events hypothesized by discourse processing into a complete template structure, deciding whether to default the value of template slots not found in the event structure (e.g, using date and location information in the header), and creating the required template forms.

A critical component for future work is a fragment combining algorithm. Based on local syntactic and semantic information, this algorithm combines fragments to provide more complete analyses of the input [Weischedel, et al., 1991a]. Though the fragment combining algorithm implemented rules for finding conjoined phrases,[2] prepositional phrase attachment[3], appositive recognition, and correcting errors made by MITFP (e.g., combining adjacent fragments into a single noun phrase), there was not time to add all of the structural semantic rules for the resulting fragments. Therefore, the combining algorithm was not thoroughly evaluated in MUC-3.

## OFFICIAL RESULTS

There are several alternative ways to run the algorithms, representing alternative degrees of conservative versus aggressive hypothesis of templates and slot values. Two alternatives produced a noticeably different tradeoff between recall on the one hand, and precision and recall on the other.

---

[1] We are now in process of integrating BBN's POST [Meteer, Schwartz, and Weischedel, 1991] probabilistic part-of-speech tagger for the tagger in MITFP. In preparing for MUC-3, creating components took priority over replacing existing components.

[2] MITFP usually produces fragments where a conjoined phrase appears because local syntactic information is usually not sufficient to reliably predict the correct parse.

[3] MITFP usually does not attach prepositional phrases because of the inherent ambiguity.

In the "max tradeoff" version, the one which produced the least difference between recall and precision, a template is produced for an event even if the event has no target nor a date identified the text. The output of the scoring program appears in Figure 2. The overall recall is 42%; precision is 52%; overgeneration is 22%.

A more conservative version produces a template only if a date and target can be found. Furthermore, information pertaining to the event (e.g., time, location, instrument, etc.) must be found within one paragraph of the phrase(s) designating the event. This is particularly interesting, since at a cost of 3 points in recall, a gain of 6 points in precision and a cut of one third in overgeneration is achieved.

## EFFORT SPENT

We estimate that roughly seven person months went into our effort. At least half of that was creating algorithms and domain-independent software, since we did not have a complete message processing system prior to this effort. About 5% of the effort went to additions to the domain-independent lexicon. Therefore the (roughly) 4 months of person effort for our first domain should not have to be repeated for a new domain.

The remaining 3 months were spent on domain-specific tasks:

- Domain-specific rules in semantics, in the discourse component, in the template generator, and in the domain model
- Domain-dependent lexical additions.

## TRAINING DATA AND TECHNIQUES

The 1300 messages of the development corpus were used at various levels as training data. PLUM was run over all 1300 messages to detect, debug and correct any causes of system breaks. The perpetrator organization slot for all 1300 messages was used to quickly add their names to the domain-dependent lexicon. After running our part-of-speech tagger (POST) over the development corpus, the statistical algorithm for predicting words of Spanish origin was run over the list of previously unknown words. Those predicted as Spanish in origin were then run through manually to add Spanish names to the lexicon.

| SLOT | POS | ACT | COR | PAR | INC | ICR | IPA | SPU | MIS | NON | REC | PRE | OVG | FAL |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MATCHED ONLY | 1466 | 1263 | 569 | 189 | 221 | 63 | 151 | 284 | 487 | 780 | 45 | 52 | 22 | |
| MATCHED/MISSING | 1485 | 1263 | 569 | 189 | 221 | 63 | 151 | 284 | 506 | 793 | 45 | 52 | 22 | |
| ALL TEMPLATES | 1485 | 2530 | 569 | 189 | 221 | 63 | 151 | 1551 | 506 | 2574 | 45 | 26 | 61 | |
| SET FILLS ONLY | 618 | 427 | 198 | 86 | 76 | 24 | 80 | 67 | 258 | 508 | 39 | 56 | 16 | 1 |

**Figure 2: "Max-tradeoff" Results**

| SLOT | POS | ACT | COR | PAR | INC | ICR | IPA | SPU | MIS | NON | REC | PRE | OVG | FAL |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MATCHED ONLY | 1360 | 1063 | 544 | 157 | 211 | 75 | 128 | 151 | 448 | 739 | 46 | 58 | 14 | |
| MATCHED/MISSING | 1465 | 1063 | 544 | 157 | 211 | 75 | 128 | 151 | 553 | 824 | 42 | 58 | 14 | |
| ALL TEMPLATES | 1465 | 1659 | 544 | 157 | 211 | 75 | 128 | 747 | 553 | 1542 | 42 | 38 | 45 | |
| SET FILLS ONLY | 609 | 372 | 194 | 66 | 72 | 31 | 64 | 40 | 277 | 529 | 37 | 61 | 11 | 0 |

**Figure 3: Official Results for Optional Condition**

A subset of the development set was used more intensively as training data. Approximately 95,000 words of text (about 20% of the development corpus) was tagged as to part of speech and labelled as to syntactic structure; that was part of the DARPA-funded TREEBANK project at the University of Pennsylvania. The bracketed text first provided us with a frequency-ranked list of head verbs, head nouns, and nominal compounds. For each of these we

added a pointer to the domain model element that is the most specific super-concept containing all things denoted by the verb, noun, or nominal compound. As mentioned earlier, the TREEBANK data was then used with the lexical relation to the domain model to hypothesize case frames for verbs.

Given a sample of text, we annotate each noun, verb, and proper noun in the sample with the semantic class corresponding to it in the domain model. For instance, *dawn* would be annotated <time>, *explode* would be <explosion event>, and *Yunguyo* would be <city>. We estimate that this semantic annotation proceeded at about 90 words/hour.

From a single example parse tree in TREEBANK, one can clearly infer that bombs can explode, or more properly, that *bomb* can be the logical subject of *explode*, that *at dawn* can modify *explode*, etc. Naturally, good generalizations based on the instances are critical, rather than the instances themselves.

Since we have a hierarchical domain model, and since the manual semantic annotation states the relationship between lexical items and concepts in the domain model, we used the domain model hierarchy as a given set of categories for generalization. However, the critical issue is selecting the right level of generalization given the set of examples in the supervised training set.

We extended and generalized a known statistical procedure (Katz, 1987) that selects the minimum level of generalization such that there is sufficient data in the training set to support discrimination of cases of attaching phrases (arguments) to their head. This is detailed in [Weischedel, Meteer, Schwartz, 1991].

The automatically hypothesized verb case frames were then reviewed manually and added to the lexicon.

Lastly, the first one hundred messages of the development corpus were used for detailed system debugging, while the 100 messages of TST1 were used as a test set to measure our progress at least once a week. Throughout, we only looked at the summary output from the scoring procedure, rather than adding to the lexicon based on TST1 or debugging the system based on particular messages.

Our performance on the hundred messages TST1 is shown in Figure 4.

# CONCLUSIONS

## Successes

PLUM has the following key features:

1. Fragment production based on the lexicon and local syntactic information
2. Partial understanding provided for each fragment found.
3. Event-based and template-based knowledge to find relations among entities when syntax/semantics cannot find them.
4. Statistical language models at multiple levels.

These were the key to PLUM's performance in MUC-3. All components of PLUM except the domain-specific knowledge bases seem transferable to other domains.

## Improvements Desired

Coverage in both the semantics and discourse components can and should be increased. The fragment combining component should be tested and evaluated thoroughly, since it was not thoroughly tested in MUC-3. Rather than a purely deterministic fragment finding algorithm as in MITFP, a fragment finding algorithm based on probabilistic language models and local search might provide more accurate prediction of phrase boundaries and phrase types.

The template generator today is based on hand-crafted rules of thumb. Within the next two years we hope to develop and test an acquisition algorithm that would acquire most of the rules from examples in a new domain.

## Lessons Learned

The degree of success obtained by marrying fragment processing/partial understanding with statistical techniques has been quite gratifying. The availability of 1300 messages with their desired templates was invaluable. Furthermore, the value of annotated text as in TREEBANK was great; the provision of more data is warranted and would be even better. It would also have been impossible to determine our progress over large sets of messages without the scoring program.

## ACKNOWLEDGMENTS

The work reported here was supported in part by the Defense Advanced Research Projects Agency and was monitored by the Rome Air Development Center under Contract Nos. F30602-87-D-0093 and F30602-91-C-0051. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the United States Government.

## REFERENCES

de Marcken, C.G. Parsing the LOB Corpus. *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics* 1990, 243-251.

Jacobs, P., Krupka, G.P., and Rau, L.F. Lexicon-Semantic Pattern Matching as a Companion to Parsing in Text Understanding, *Proceedings of the Fourth DARPA Workshop on Speech and Natural Language*, February 1991.

Katz, S.M. Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-35 No. 3, March 1987.

Meteer, M., Schwartz, R., and Weischedel, R. Empirical Studies in Part of Speech Labelling., *Proceedings of the Fourth DARPA Workshop on Speech and Natural Language* , February 1991.

Weischedel, R., Ayuso, D.M., Bobrow, R., Boisen, S., Ingria, R., and Palmucci, J., Partial Parsing, A Report on Work in Progress, *Proceedings of the Fourth DARPA Workshop on Speech and Natural Language*, 1991a.

Weischedel, R., Meteer, M., and Schwartz, Applications of Statistical Language Modelling to Natural Language Processing, unpublished manuscript, 1991b.

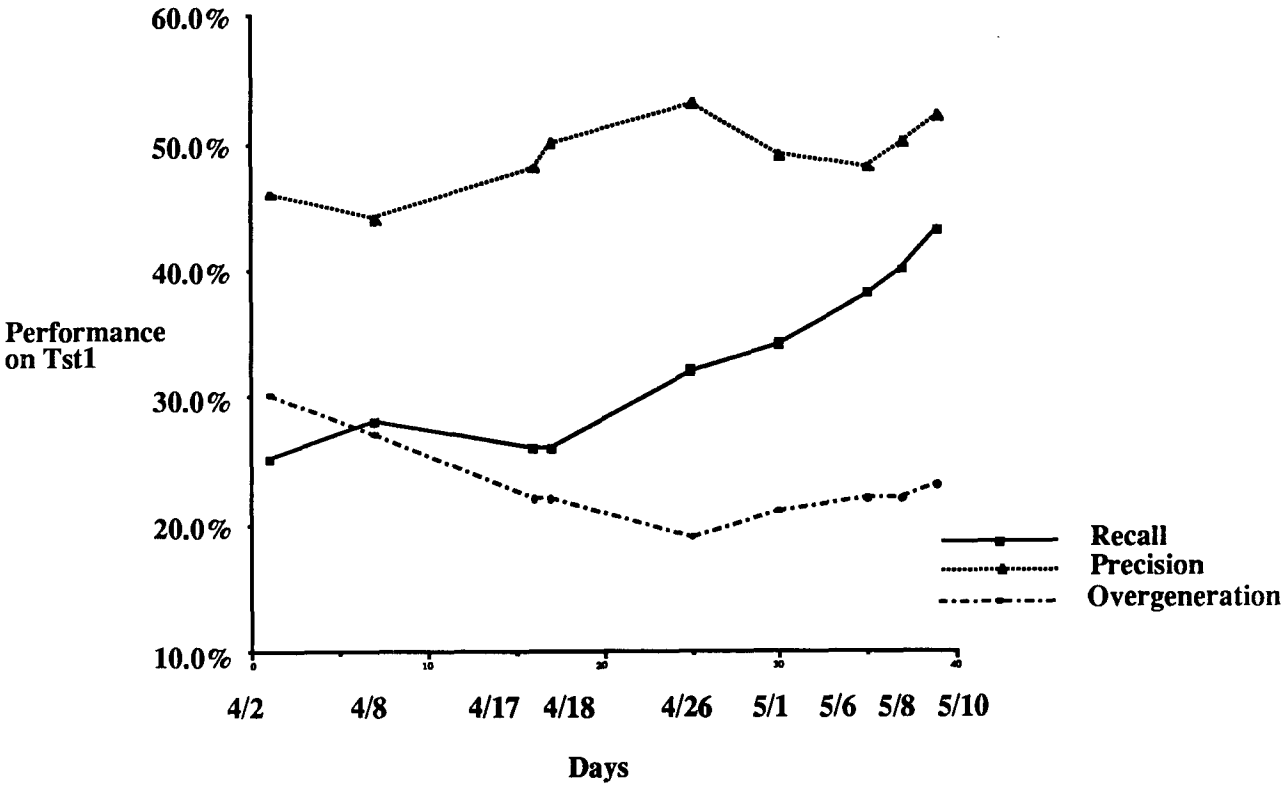


**Figure 4: TST1 performance over development period**