SRA SOLOMON:
MUC-4 TEST RESULTS AND ANALYSIS

Chinatsu Aone, Doug McKee, Sandy Shinn, Hatte Blejer

Systems Research and Applications (SRA)
2000 15th Street North
Arlington, VA 22201
aonec@sra.com

INTRODUCTION

In this paper, we report SRA's results on the MUC-4 task and describe how we trained our natural language processing system for MUC-4. We also report on what worked, what didn't work, and lessons learned. Our MUC-4 system embeds the SOLOMON knowledge-based NLP shell which is designed for both *domain-independence* and *language-independence*. We are currently using SOLOMON for a Spanish and Japanese text understanding project in a different domain. Although this was our first year participating in MUC, we have built and are currently building other data extraction systems.

RESULTS

Our TST3 and TST4 results are shown in Figures 1 and 2. The similarity of these scores as well as their similarity to SRA-internal testing results reflects the portability of SRA's MUC-4 system. In fact, our score on the TST4 texts was better than that of TST3, even though those texts covered a different time period than that of the training texts or TST3.

Our matched-only precision and recall for both test sets were very high (TST3: 68/47, TST4: 73/49). When SOLOMON recognized a MUC event, it did a very accurate and complete job at filling the requisite templates.

SOLOMON performance was tuned so that the all-templates recall and precision were as close as possible to maximize the F-Measure. As shown in Figure 3, our F-Measure steadily increased over time. The fact that this slope has not yet leveled off shows SOLOMON's potential for improvement.

EFFORT SPENT

We spent a total of 9 staff months starting January 1, 1992 through May 31, 1992 on MUC-4. A task-specific breakdown of effort is shown in Figure 4. The bulk of the work was spent porting SOLOMON to a new domain with new vocabulary, concepts, template-output format, and fill rules. Approximately 72% of the effort was domain-dependent. However, about 63% of the total effort was *language-independent*, i.e. it would be directly applicable to understanding texts about terrorism in *any* language. We expect that our English MUC-4 system could be ported to a new language in about 3 months, given a basic grammar, lexicon and preprocessing data similar to the ones which existed for English. We partially demonstrated this

	REC	PRE	OVG	FAL
MATCHED/MISSING	27	68	8	
MATCHED/SPURIOUS	47	32	57	
MATCHED ONLY	47	68	8	
ALL TEMPLATES	27	32	57	
TEXT FILTERING	71	85	15	23

	P&R	2P&R	P&2R
F-MEASURES	29.29	30.86	27.87

Figure 1: TST3 Results

	REC	PRE	OVG	FAL
MATCHED/MISSING	38	73	4	
MATCHED/SPURIOUS	49	31	59	
MATCHED ONLY	49	73	4	
ALL TEMPLATES	38	31	59	
TEXT FILTERING	91	75	25	35

	P&R	2P&R	P&2R
F-MEASURES	34.14	32.19	36.36

Figure 2: TST4 Results

claim by showing our MUC-4 system processing English, Japanese and Spanish newspaper articles about the murder of Jesuit priests at the demonstration session of MUC-4. We spent less than 2 weeks after the final test adding MUC-specific words to Spanish and Japanese lexicons, and extending the grammars of the two languages.

Data

40% of the total effort building MUC-data was spent on lexicon and KB entry acquisition. Much of this data was acquired automatically. We used the supplied geographical data to automatically build location lexicons and KBs. Using the development templates, we acquired lexical and KB entries for classes of domain terms such as human and physical targets and terrorist organizations. We automatically derived subcategorization information for the domain verbs from the development texts (cf. [1]). These automatically acquired lexicons and KBs did require some manual cleanup and correction.

Certain multi-word phenomena which occur frequently in texts but are unsuitable for general parsing were handled by pattern matching during Preprocessing. For example, we created patterns for Spanish phrases, complex location phrases, relative times, and names of political, military and terrorist organizations.

Modifications to SOLOMON's broad-coverage English grammar included adding more semantic restrictions, extending some phrase-structure rules, and improving general robustness.

Based on our knowledge engineering effort, we built a set of commonsense reasoning rules that are described in detail in our system description. Our EXTRACT module recognizes MUC-relevant events in the output of SOLOMON and translates them into MUC-4 filled templates. We implemented all the domain-specific information as mapping rules or simple conversion functions (e.g. numeric values like "at least 5" means "5-"). This data is stored in the knowledge base, and is completely language independent.

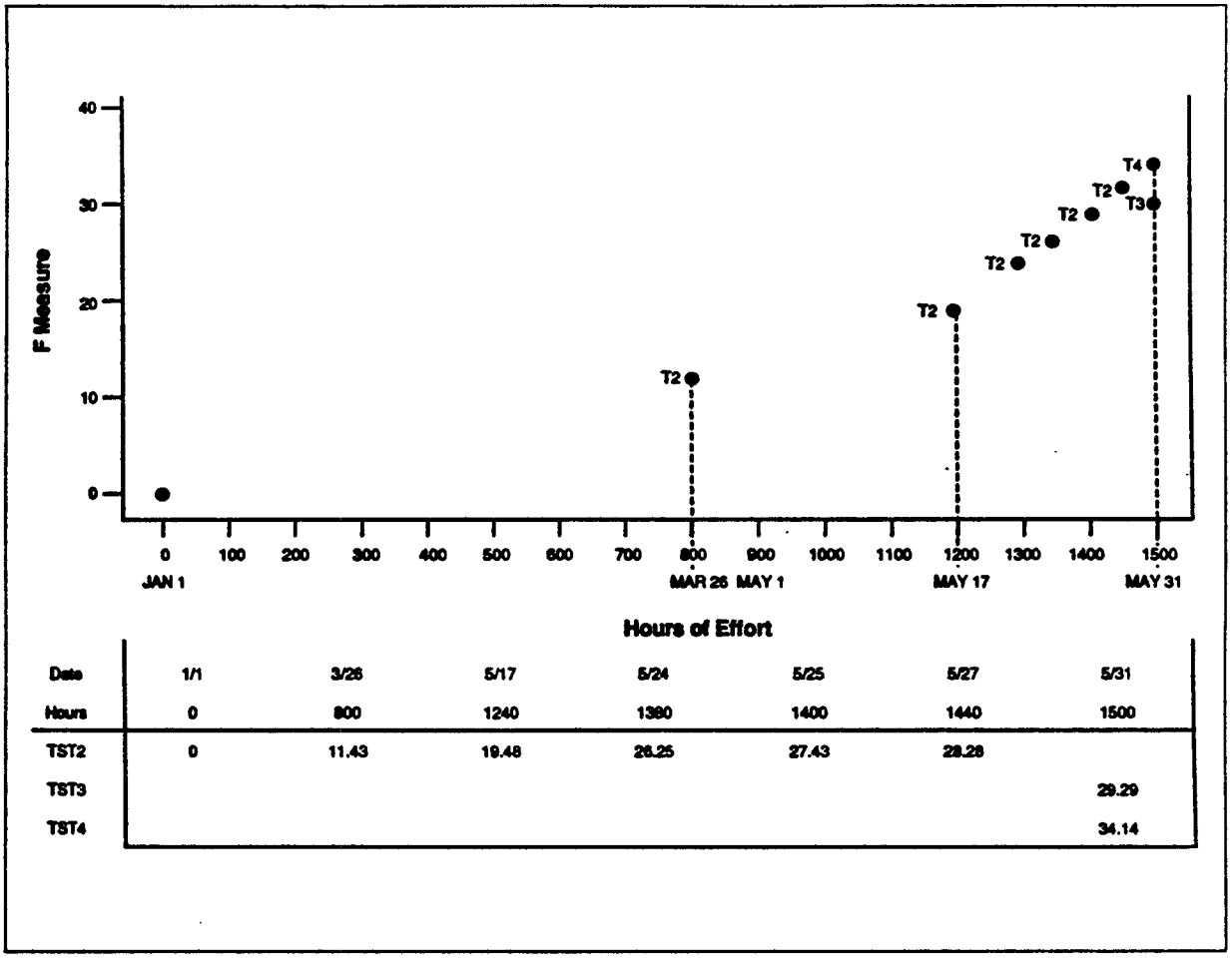

Figure 3: Tracking SOLOMON Performance

Task Category	% of Total Effort
DATA	71
Knowledge Engineering	13
Data Acquisition	30
Grammar	7
Pragmatic Inference Rules	11
Extract Data	10
PROCESSING	29
Message Zoning	3
Extract Extensions	7
Testing	10
Misc. Bug Fixing	10

Figure 4: Breakdown of Effort Spent for MUC-4

Processing

We spent 1 week porting our existing Message Zoner to deal with message headers in MUC messages. The Message Zoner could already recognize more general message structures such as paragraphs and sentences. We extended EXTRACT while maintaining domain and language independence of the module. Features added included event merging and handling of flat MUC templates instead of the more object-oriented database records that SOLOMON is accustomed to. Our time spent on fixing bugs was distributed throughout the system, but problems in Debris Parsing and Debris Semantics received the most attention.

SYSTEM TRAINING

We used TST2 texts for blind testing and the entire 1300 development texts for both testing and training material. The development set was crucial to both our automated data acquisition and our knowledge engineering task. We performed frequent testing to track and direct our progress. To raise recall, we focussed on data acquisition; to raise precision, we focussed on stricter definitions of "legal" MUC events. To improve overall performance, we focussed on more robust syntactic and semantic analysis and more reliable event merging.

LIMITING FACTORS

The two main limiting factors were the number of development texts and templates and the amount of time allotted for the MUC-4 effort. With more texts, we could have applied other more data-intensive automated acquisition techniques and had more examples of phenomena to draw upon. With more time, we would add more domain-dependent lexical knowledge and additional pragmatic inference rules. We also need to tune our EXTRACT mapping rules more finely and improve our discourse module for both NP reference and event reference resolution. Integration of existing on-line resources such as machine-readable dictionaries, the World Factbook, or WordNet would also improve system performance. A more extensive testing and evaluation strategy at both the blackbox and glassbox levels would help direct progress, but was not feasible in the amount of time we had.

WHAT WAS OR WAS NOT SUCCESSFUL

There were several areas where *hybrid* solutions worked very well. Totally automated knowledge acquisition was quite successful when supplemented by manual checking and editing of domain-crucial information. Similarly, augmenting a pure bottom-up parser with "simulated top-down parsing" (See SRA's MUC-4 System Description) worked well. Improved Debris Semantics and significantly extended Pragmatic Inferencing were also important contributors to the system's performance.

REUSABILITY

SRA's SOLOMON NLP system has been designed for portability and proven to be highly reusable. This includes portability to other domains, other languages, and other applications. As shown in Figure 5, a large

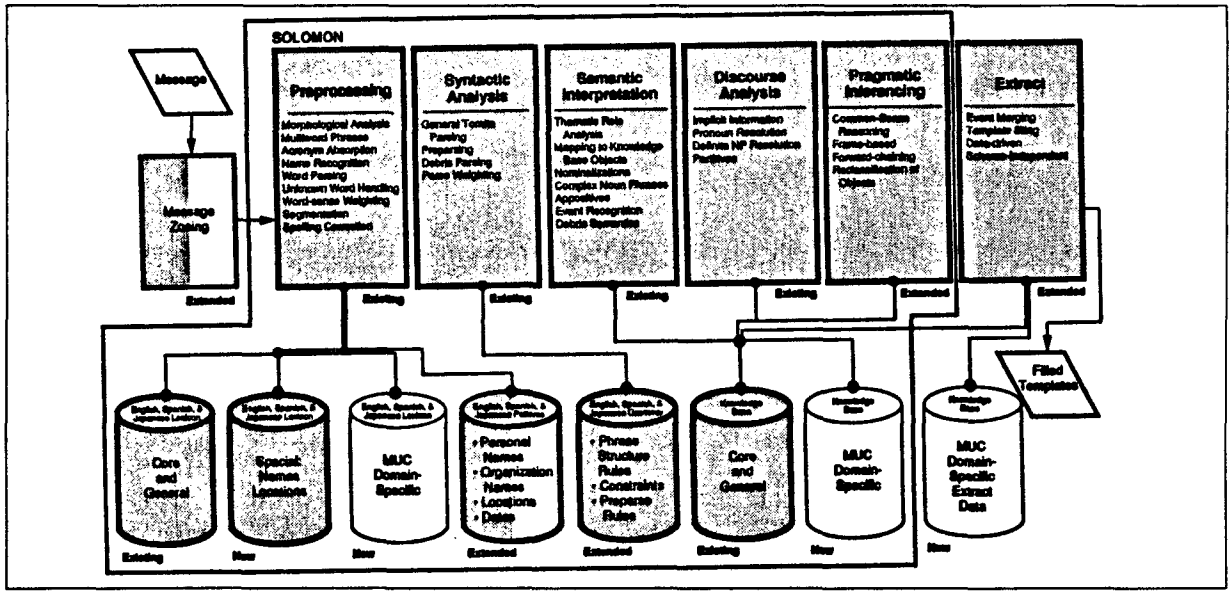

Figure 5: MUC NLP System Reusability

part of SOLOMON's data and almost all of the processing modules are completely reusable for NLP in other domains or languages.

Currently, our Spanish and Japanese data extraction project MURASAKI is using, without modification, the same processing modules and the core knowledge base as those used for MUC-4. The MURASAKI system processes Spanish and Japanese language newspaper and journal articles as well as TV transcripts. This project's domain is the AIDS disease. Thus, the only difference between our MUC-4 system and MURASAKI system is that the latter uses Spanish and Japanese lexicons, patterns and grammars, and MURASAKI domain-dependent knowledge bases. SOLOMON has also been embedded in several English message understanding systems: ALEXIS (operational) and WARBUCKS.

LESSONS LEARNED AND REAFFIRMED BY MUC-4

We have learned and reaffirmed the following points as the most crucial aspects of successful text understanding for data extraction.

Overcoming the Knowledge Acquisition Bottleneck: We must develop techniques and tools for acquiring timely, complete, and proven system data.

Solving the Parsing Problem: We need more robust, semantically constrained syntactic analysis. Grammars must be broad-coverage and highly accurate on complex input.

Developing Sophisticated Discourse Analysis: We must handle real world discourse phenomena found in actual texts. The discourse architecture must be flexible enough to accommodate particular discourse phenomena which are crucial in particular domains or languages.

MUC-4 has reaffirmed our knowledge of what is involved in porting an NLP system to a new domain. 9 staff months is a bare minimum for such an effort. Improved knowledge acquisition tools as well as

on-line resources are desirable. To ensure good results, it is necessary to have sufficient time for knowledge engineering, testing and evaluation. Our experience underscores the fact that natural language understanding is a highly data-driven problem. The system's performance is often proportional to the level of understanding of the input and output. The MUC-4 development texts and templates were extremely helpful in this regard.

References

[1] Doug McKee and John Maloney. Using Statistics Gained from Corpora in a Knowledge-Based NLP System. In *Proceedings of The AAAI Workshop on Statistically-Based NLP Techniques*, 1992.