GTE'S TEXT INTERPRETATION AID (TIA): MUC-3 TEST RESULTS AND ANALYSIS

Charles T. Taylor, II

GTE Government Systems
100 Ferguson Drive
Mountain View, California, 94039
(415) 966-2656
Email: taylorc%gtewd.dnet@gte.com

INTRODUCTION

GTE has actively participated in the Third Annual Message Understanding Conference (MUC-3) sponsored by the Naval Ocean System Center (NOSC) for the past 7 months using its natural language processing system, TIA (Text Interpretation Aid)[5]. During this period, TIA system development consisted of two tasks: (1) lexicon tool development, and (2) tailoring TIA to meet the specific needs of the MUC-3 domain, i.e., terrorism in Latin American countries. Lexicon tool development facilitates system (lexicon) adaptation to new domains (terrorism, drug interdiction, third world launchings, etc.) by semi-automating the manual task of entering words/phrases into TIA's lexicons. Tailoring the existing TIA system to parse and extract MUC-3 domain messages, has allowed GTE to participate in the MUC-3 conference tasks.

The purpose of this paper is to report our techniques and findings during these 7 months of MUC-TIA system development. Additionally, GTE's official scores for Phase I and Phase II of MUC-3 will be disclosed and discussed. The remainder of this paper is divided into seven sections describing scoring results, justification and analysis of scores, system development effort, limiting factors, training, reusability of the resultant MUC-TIA system, and final conclusions.

SCORING RESULTS

The following section discusses MUC-TIA's evaluation scores collected during the final week of system development/testing. All scores were gathered by running MUC-TIA under normal operational mode, i.e., no tradeoff testing configuration switches are utilized for optimizing testing parameters, e.g., recall vs. precision, precision vs. overgeneration, etc. MUC-TIA operates under the direct assumption of maximized recall and precision, and minimized overgeneration and fallout. For a detailed discussion of the scoring metrics, see the MUC-3 Scoring System User Manual [3]. Tables 1.0, 2.0, and 3.0 display GTE's scores for the MUC-3 NL evaluation task.

Recall

Recall is a maximized scoring metric which measures the amount of data extracted from messages and inserted into message templates during the parsing and extraction processes. During Phase II of MUC-3, overall recall (REC) for NOSC's test set "tst2-muc3" was computed to be 28% for "Matched Only"[1], 11% for "Matched/Missing", and 11% for "All Templates." However, these results at first glance seemed inconsistent with our Phase I results from NOSC's test set "tst1-muc3", shown in Table 2.0, where TIA achieved a recall of 21%[2], suggesting that recall decreased after Phase II development. To form a baseline for comparisons, GTE rescored NOSC's test set "tst1-muc3" using Phase II's scoring software. These results are shown in Table 3.0. Unfortunately, Phase II

[1] "Matched Only" refers to the totals for templates which are matched, i.e., scores are not penalized for missing or spurious slot fillers (template slot id is an exception to this rule). "Matched/Missing" contains the totals for templates which are matched, however scores are penalized for missing, but not spurious, slot fillers. "All Templates" contain totals for templates, however penalizations occur for missing and spurious slot fillers. "Set Fills Only" contains the totals for only the slots filled from a finite set.

[2] The scoring software used during Phase I of MUC-3 has been significantly modified to capture more precise scoring metrics. Phase I Grand Totals roughly correspond to Phase II's Matched/Missing template scores.

discouraging results were confirmed after rescoring tst1-muc3 when recall decreased from 31% for "Matched Only" (tst1-muc3) to 28% (tst2-muc3). "Matched Missing" and "All Templates" were consistent with scores of 11% for tst1-muc3 and tst2-muc3. This decrease may simply indicate tst2-muc3 is a more difficult message corpus to understand.

Precision

One interesting score consistent throughout the entire MUC-3 evaluation task (Phase I and Phase II) was *precision*. Precision (PRE) measures the correctness of the information extracted from the messages and placed in the templates during the parsing processes. The overall goal is to maximize precision. GTE's precision (for "tst2-muc3") was 43% for "Matched Only", 43% for "Matched/Missing" and 25% for "All Templates." After examining Phase I scores, precision did increase (although not significantly) 1%. Moreover, the rescored "tst1-muc3" precision was 42% for "Matched Only", 42% for "Matched/Missing", and 18% for "All Templates".

OverGeneration

Overgeneration is the scoring metric which measures extraneous template fills, i.e., the percentage of templates which were incorrectly spawned during the parsing and extraction processes. This metric should be minimized. GTE scored 33% for tst2-muc3 "Matched Only", 33% for "Matched/Missing", and 61% for "All Templates". During Phase I testing, GTE scored 29% overgeneration. After rescoring tst1-muc3 (after Phase II development) overgeneration increased to 35% for "Matched Only", 35% for "Matched/Missing", and 72% for "All Templates". Overgeneration slightly increased by Phase II development.

SLOT	POS	ACT	COR	PAR	INC	SPU	MIS	REC	PRE	OVG
template-id	109	84	40	0	0	44	69	37	48	52
incident-date	105	34	22	6	6	0	71	24	74	0
incident-type	109	40	24	14	2	0	69	28	78	0
category	77	40	15	0	13	12	49	19	38	30
indiv-perps	93	9	1	0	4	4	88	1	11	44
org-perps	62	4	2	0	1	1	59	3	50	25
perp-confidence	62	40	0	8	11	21	43	6	10	52
phys-target-ids	56	8	0	1	2	5	53	1	6	62
phys-target-num	38	19	2	0	2	15	34	5	10	79
phys-target-types	56	2	0	0	0	2	56	0	0	100
human-target-ids	134	10	1	3	3	3	127	2	25	30
human-target-num	87	21	10	0	10	1	67	11	48	5
human-target-types	134	2	0	0	2	0	132	0	0	0
target-nationality	16	1	0	0	0	1	16	0	0	100
instrument-types	25	0	0	0	0	0	25	0	*	*
incident-location	109	22	11	5	6	0	87	12	61	0
phys-effects	36	3	0	1	0	2	35	1	17	67
human-effects	56	3	0	1	0	2	55	1	17	67
MATCHED ONLY	526	342	128	39	62	113	297	28	43	33
MATCHED/MISSING	1364	342	128	39	62	113	1135	11	43	33
ALL TEMPLATES	1364	593	128	39	62	364	1135	11	25	61
SET FILLS ONLY	571	131	39	24	28	40	480	9	39	30

Table 1.0: Official TST2-MUC3 Phase II Scores

SLOT	POS	ACT	COR	PAR	INC	SPU	MIS	REC	PRE	OVG
template-id	95	135	59	0	0	76	36	62	44	56
incident-date	92	56	15	25	16	0	36	30	49	0
incident-type	95	59	36	1	22	0	36	38	62	0
category	66	59	29	0	12	18	25	44	49	30
indiv-perps	87	14	1	1	11	1	74	2	11	7
org-perps	58	15	7	3	2	3	46	15	57	20
perp-confidence	98	59	28	2	20	9	48	30	49	15
phys-target-ids	52	0	0	0	0	0	52	0	100	0
phys-target-num	40	26	2	0	4	20	34	5	8	77
phys-target-types	47	0	0	0	0	0	47	0	100	0
human-target-ids	94	32	2	4	17	9	71	4	12	28

human-target-num	68	32	12	0	15	5	41	18	38	16
human-target-types	76	31	15	1	8	7	52	20	50	22
target-nationality	23	0	0	0	0	0	23	0	100	0
instrument-types	17	0	0	0	0	0	17	0	100	0
incident-location	95	32	8	12	12	0	63	15	44	0
phys-effects	29	0	0	0	0	0	29	0	100	0
human-effects	29	30	2	2	6	20	19	10	10	67
GRAND TOTAL	1161	580	216	51	145	168	749	21	42	29

Table 2.0: TST1-MUC3 after Phase I development

SLOT	POS	ACT	COR	PAR	INC	SPU	MIS	REC	PRE	OVG
template-id	91	85	27	0	0	58	64	30	32	68
incident-date	88	25	12	9	4	0	63	19	66	0
incident-type	91	27	18	0	9	0	64	20	67	0
category	62	27	10	0	7	10	45	16	37	37
indiv-perps	82	14	4	0	5	5	73	5	28	36
org-perps	57	5	3	0	0	2	54	5	60	40
perp-confidence	94	27	14	3	6	4	71	16	57	15
phys-target-ids	52	8	4	0	0	4	48	8	50	50
phys-target-num	40	17	2	0	3	12	35	5	12	70
phys-target-types	47	5	4	0	0	1	43	8	80	20
human-target-ids	89	2	1	0	1	0	87	1	50	0
human-target-num	64	12	1	0	9	2	54	2	8	17
human-target-types	72	2	0	1	1	0	70	1	25	0
target-nationality	23	2	0	0	2	0	21	0	0	0
instrument-types	17	0	0	0	0	0	17	0	*	*
incident-location	91	22	9	6	7	0	69	13	54	0
phys-effects	29	3	1	1	1	0	26	5	50	0
human-effects	28	1	0	0	0	1	28	0	0	100
MATCHED ONLY	382	284	110	20	55	99	197	31	42	35
MATCHED/MISSING	1117	284	110	20	55	99	932	11	42	35
ALL TEMPLATES	1117	665	110	20	55	480	932	11	18	72
SET FILLS ONLY	463	94	47	5	26	16	385	11	53	17

Table 3.0: TST1-MUC3 rescored after Phase II development

JUSTIFICATION AND ANALYSIS OF SCORES

Although the above stated scores seem rather discouraging or low, there are several valid justifications for such occurrences. The sections which follow explain each of the justifications.

System Defaults Less

Phase I scores left GTE with some artificial results for recall and precision. Several slot fillers were the direct result of system defaults. This in turn filled many slots with correct fillers, but for wrong reasons, a phenomenon which Grishman calls "uncoupling input and output." For example, during Phase I scoring, the MUC-TIA system defaulted the template slot Perpetrator: Confidence to the set list filler of "REPORTED AS FACT"; however, no real analysis was performed. Since "REPORTED AS FACT" was the most used correct slot filler, the score was artificially inflated.

Backend Translation

The MUC-TIA System's internal semantic representation of a parse consists of realizations of structured concepts. Structured concepts are frame-like knowledge representations which maintain slot fillers. During the semantic parsing process structured concepts are realized (essentially instantiated) by slot fillers such as simple text strings, or with more complex fillers such as demons, which are spawned. For example, an event such as a bombing instantiates a structured concept bombing-p with slots for actor (who performed the bombing), theme (what was bombed), location (where the bombing took place), etc. These realized structured concepts in turn represent the message parsed and maintain the data extracted. A backend translation process then maps and normalizes the data maintained in the structured concepts and places it in the appropriate templates.

This domain specific back-end translation module has not been fully tested and/or implemented. Many formatting issues still need to be resolved. Moreover, template merging techniques/heuristics are still being tested to determine optimal methods. Additionally, complete slot cross-referencing has not been completed and fully tested. As a result, many incorrect and partial matches occurred during the scoring process, thereby causing a detrimental effect on GTE's scores. Although the correct data was extracted from the message and maintained in the system's internal representation, i.e., structured concepts, the actual template slot was filled incorrectly due to the back-end translation process. For example, message TST2-MUC3-0034's HUMAN TARGET: TYPE correct slot filler is: POLITICAL FIGURE: "JECAR NEGHME", however, TIA's response template indicates "SPOKESMAN": "-". After further review of TIA's internal representation of the message, a murder-p structured concept was properly instantiated with ,"JECAR NEGHME", a SPOKESMAN for the MIR, thereby properly identifying the appropriate human target.

GTE has identified these "data extraction" problems with the back-end translator and recommends this module be rewritten.

New Semantics Partially Implemented

During Phase II development several new semantic ideas were implemented which were not fully tested. For instance, to assist in filling the PERPETRATOR: CONFIDENCE slot, a "mode-p" prediction prototype [1] was defined which maintains two slots: By-Whom-S, and Insert-Mode-S. The By-Whom-S slot is filled by the *authoritative* figure which is found in the last act (this prediction is defined in the mode-p prediction prototype's control structure.) The "insert-mode-s" slot's purpose is to inhibit the generation of a new template. For example, message TST2-MUC3-0011 states

> "The chief of the armed forces joint chiefs of staff have categorically **denied** that there are any **rifts** between Salvadoran army officers and U.S. military, as asserted by the Washington Post."

Normally, the word *rifts* spawns a realization of an attack template; however, the phase, *denied that* inhibited the attack template. This experimental mechanism has not been fully tested.

Time of Domain Specific System Development

During the MUC-3 development period, several lexicon tools have been implemented which facilitate development for new domains, e.g., terrorism, drug interdiction, third world launches, etc. These semi-automatic tools allow the lexicon developer to browse the message corpus and define lexical entries through a series of menus[3]. Additionally, sorting utilities were developed which operate on the automatically defined lexical entries. These tools are imperative to training any natural language processing system to a new domain. These tools have greatly increased the lexicon developers productivity while reducing debugging time. Since the majority of MUC-3's development time was devoted towards tool implementation, a minimal amount of MUC-3 domain-specific system development was performed, which is reflected in GTE's scores.

SYSTEM DEVELOPMENT EFFORT

The majority of the MUC-3 system development effort involved lexicon development issues (discussed below). The construction of lexicon development tools and macros absorbed the majority of the development time. Approximately 200 of the 360 hours of system development were devoted towards these tasks. The balance, approximately 160 hours, were devoted towards actual MUC-3 task specific system development. As a result, GTE's scores were adversely affected.

LIMITING FACTORS

The following sections describe some of the limiting factors and problems which GTE had to overcome in order to participate in the MUC-3 Project.

[3] This menu approach will be modified and a human machine interface using X11 and Motif will be implemented for the lexicon development tools in the near future.

Person Resources

GTE has devoted two software engineers working on the MUC-3 Project for varied amounts of time. One software engineer (employed by GTE for six years) worked on the original TIA system first established in 1985. During Phase II development, he devoted approximately 80 hours to MUC-3 domain specific tasks. The other software engineer (employed by GTE for approximately one year) devoted approximately 280 hours towards lexicon tool development, system administration (Sun 4/490 Sparc Server), MUC-3 domain specific system development, scoring and interpreting results. As a result, GTE was not able to consecrate the desired time to MUC-3 (domain specific) system development.

Syntactic Parser's Combinatorial Explosion Problem

A second limiting factor which arose and was eventually solved was the syntactic parser's combinatorial explosion problem. This problem occurred due to the top-down exhaustive nature of the parser. The problem originally became apparent when several non-terminal syntactic constituents, e.g., regions, organizations, became extremely large and unwieldy. Since the parser expands non-terminals in a uniform, non-heuristic manner, all applicable grammar rules are fired - even rules which are not viable. For example, if two rules present in the syntactic grammar are of the form:

```
Rule 1:       <A>   -->   <B> <C>
Rule 2:       <A>   -->   <B> <D>
```

and Rule 1 fails because <B> cannot be expanded during the parse, Rule 2 or any other rule of the form: <A> --> <B> was still attempted to be expanded, even though it cannot yield any positive results. Consider the following dev-muc3 excerpt (labeled "Failed String") and the <Name-Position> syntactic rules shown below.

```
Failed String:       "Alfredo Cristiani, president of El Salvador"
<Name-Position> --> <Name> <comma> <Region> <Position>   |
                    <Name> <comma> <Region> <apostrophe-s><Position>
```

Since the string's parse fails at the non-terminal <Region> in the first <Name-Position> rule (because *president* cannot be a <Region>), the parse should not be permitted to try parsing using the second option of <Name-Position>. When the number of nonterminal expansions for a single nonterminal is "small", this issue is not problematic. However, as the number of expansions becomes "large", the inefficiency degrades the parser dramatically.

The problem was solved by establishing/marking the set of non-terminals which may contain a large number of expansions, and maintaining failed parse states within the current phrases parse. If the current phrase being parsed is in a state which has failed at some prior time and the current nonterminal being expanded is "large", the system does not try to expand the current nonterminal using the current rule. This pruning of the search space does not alter the language recognized, i.e., all previously parsable constructs are still viable and are parsed appropriately.

This solution caused dramatic results during several parses. Prior to this optimization, a sample parse of a phrase containing approximately three words which yield "large" nonterminals took the MUC-TIA system approximately 145 CPU seconds to run. After the optimization was implemented, the same phrase took approximately 0.4 CPU seconds - obviously a worthwhile improvement.

TRAINING

As previously discussed, one MUC-TIA training task consisted of automating the process of lexicon development. GTE has developed two tools and several domain specific macros to train the system, each discussed below in more detail.

Lexicon Learner and Sorter Tools

The lexicon learner tool/utility automates the process of entering unknown (essentially undefined) words and/or phrases into the appropriate syntactic lexicon with the appropriate syntactic and semantic features. Consider the following excerpt from one of the dev-muc3 messages.

```
"Ricardo Alfonso Castellar, Mayor of (Achi.UNKNOWN), in the Northern
Department of Bolivar, who was kidnapped on 5 January, apparently by Army
```

```
of National Liberations (ELN) guerillas, was found (slaughtered.UNKNOWN)
today, according to authorities."
```

When the lexicon learner encounters the unknown lexical entry "*Achi*", the system prompts for the appropriate syntactic and semantic information necessary to sufficiently define the lexical entry as shown below. The city Achi is defined by a `Def-Region` macro which maintains fields for grammar, syntax, part-of, and type. The `grammar` field is initialized to `muc3` (the grammar for the MUC-3 project), `syntax` (specifies the list of possible articulations for the lexical entry) is set to the list consisting of one element, `(achi)`, `part-of` (specifies the region's hierarchical constituents) is set to `bolivar`, and the `type` field (specifies the region's demography, e.g., village, city, state, country, continent, etc.) is set to `city`.

```
(Def-Region Achi
       :grammar muc3
       :syntax (achi)
       :part-of bolivar
       :type city)
```

As the defining process continues, the lexicon learner will encounter the second unknown lexical entry "*slaughtered*", prompt for the appropriate information and then construct the following lexical entry:

```
(Def-Event slaughtered
       :grammar muc3
       :syntax (slaughtered)
       :predicts ((murder-p
                      input-text-s (PT-TO-STR))))
```

These two lexical entries are then appended to the appropriate lexicon file and are compiled into the MUC-TIA system during the next *Make* of the parser.

The lexicon learner was run on approximately 750 dev-muc3 messages over a period of approximately one month. In that time, the MUC-TIA system lexicon grew from approximately 2000 lexical entries to over 25,000 lexical entries. During training development all 1200 dev-muc messages could have been run through the lexicon learner. However, due to time constraints and new-word vs. training time return, GTE software engineers elected not to continue with the lexicon learning.

Actual system development took place on approximately 3 dev-muc3 messages. Once again this training statistic occurred due to time and budgetary constraints. GTE plans to continue development in this domain in expectation of participating in MUC-4 next year.

Domain Specific Macros

The second type of lexicon development facility which was implemented for the MUC-3 project was a series of specialized macros which facilitate the definition of regions, events, people, organizations, terrorist groups, last names, etc. Each macro performs its unique job by establishing the grammar, syntax, and several macro dependent specialized fields. For example, `def-region` maintains `part-of`, `type`, and `predicts` fields. Moreover, the lexical entry "slaughtered" defined above establishes a `predicts` field of `murder-p`. This `predicts` field may trigger an instantiation (not necessarily a realization) of the `murder-p` structured concept.

REUSABILITY

The majority of the MUC-TIA System may be reusable for other terrorism domains; however, should an entirely new domain be needed (such as third world launches or aircraft tracking), approximately 75% of the lexicon would need to be replaced. This task is not as insurmountable as it once was (pre-MUC-3) due to the lexicon tools developed during the MUC-3 project.

Additionally, since the back-end translator is very domain specific, a rewrite for the new domain would be necessary to adapt a new template structure.

CONCLUSIONS

Although GTE's raw scores seem rather discouraging, we feel significant progress has been made in an effort to solve the complex natural language/text interpretation problems posed by the MUC-3 Project. Encouraging factors/developments such as: TIA's speed, expandability, lexicon development tools, and engineering experience gained through the MUC-3 effort have positioned GTE in the right direction so that future research and development efforts will succeed not only in the MUC-3 terrorism domain, but in any domain which needs natural language/text interpretation technologies. We look forward to participating in next year's MUC-4.

ACKNOWLEDGEMENTS

GTE would like to thank Beth Sundheim of NOSC for her efforts during MUC-3. Without her help, many technical and managerial issues/problems would not have been solved. Additionally, GTE would like to thank Charles Wayne of DARPA for providing financial support and allowing GTE to participate in MUC-3.

The author would like to thank Robert Dietz of GTE for providing technical support during the MUC-TIA development efforts, and Carl Caley and Ken Duvall of GTE for providing the administrative assistance throughout the entire MUC-3 effort.

REFERENCES

[1] Dietz, R., "Description of the GTE TIA System Used for MUC-3", Proceedings of the Third Annual Message Understanding Conference (MUC-3), Naval Ocean Systems Center (NOSC), San Diego, CA, May 21-23, 1991.

[2] Dietz, R., "Software Design Document for Long Range Cruise Missile Analysis and Warning System (LAWS) and Text Interpretation Aid (TIA)", GTE Government Systems, August, 1990.

[3] Haverford, P., "The MUC-3 Scoring Software User Manual", GE Corporate Research and Development, April, 1991.

[4] Taylor, C., Dietz, R., "Adapting a Natural Language System for Large, Less-Constrained Domains", The Eighth Annual Advanced Military Intelligence Conference, Greenbelt, Maryland, March 12-15, 1991.

[5] Taylor, C., "MUC-3 Phase I at GTE Government Systems", Advanced Decision Systems (ADS), Mountain View, CA, February 12-14, 1991.