HUGHES RESEARCH LABORATORIES TRAINABLE TEXT SKIMMER: MUC-4 TEST RESULTS AND ANALYSIS

Stephanie E. August
Hughes Aircraft Company
Electro-Optical and Data Systems Group
P.O. Box 902 -- EO E52 C235
El Segundo, CA 90245-0902
august@sed170.hac.com
(310) 616-6491

Charles P. Dolan
Hughes Research Laboratories
3011 Malibu Canyon Road M/S RL96
Malibu, CA 90265
cpd@aic.hrl.hac.com
(310) 317-5675

SUMMARY OF MUC-4 PERFORMANCE

Table 1 shows the official template-by-template score results for the Hughes Trainable Text Skimmer used for MUC-4 (TTS-MUC4) on TST3. TTS is a largely statistical system, using a set of Bayesian classifiers with the output of a shallow parser as features. (See the System Summary section of this volume for a detailed description of TTS-MUC4).

SLOT	POS	ACT	COR	PAR	INC	ICR	IPA	SPU	MIS	NON	REC	PRE	OVG	FAL
template-id	112	106	63	0	0	0	0	43	49	0	56	59	40	
inc-date	109	101	22	15	24	22	15	40	48	6	27	29	40	
inc-loc	112	87	11	39	4	0	17	33	58	10	27	35	38	
inc-type	112	106	55	8	0	0	0	43	49	0	53	56	40	4
inc-stage	112	106	59	0	4	0	0	43	49	0	53	56	40	13
inc-instr-id	33	14	5	1	0	1	1	8	27	127	17	39	57	
inc-instr-type	52	14	4	0	2	0	0	8	46	109	8	28	57	0
perp-inc-cat	69	101	28	0	10	0	0	63	31	23	40	28	62	30
perp-ind-id	85	87	12	5	19	2	5	51	49	35	17	17	59	
perp-org-id	52	52	12	0	7	1	0	33	33	72	23	23	63	
perp-org-conf	52	52	4	2	13	0	2	33	33	72	10	10	63	5
phys-tgt-id	66	112	13	2	10	0	2	87	41	74	21	12	78	
phys-tgt-type	66	112	10	4	11	0	3	87	41	74	18	11	78	4
phys-tgt-num	67	122	13	7	5	0	7	97	42	74	25	14	80	
phys-tgt-nation	2	0	0	0	0	0	0	0	2	154	0	*	*	0
phys-tgt-effect	39	112	6	6	2	0	5	98	25	82	23	8	88	10
phys-tgt-total-num	0	39	0	0	0	0	0	39	0	116	*	0	100	
hum-tgt-name	57	173	22	5	9	1	5	137	21	68	43	14	79	
hum-tgt-desc	132	222	29	24	17	1	24	152	62	35	31	18	68	
hum-tgt-type	146	371	35	16	32	1	13	288	63	23	29	12	78	17
hum-tgt-num	146	389	35	32	16	1	26	306	63	23	35	13	79	
hum-tgt-nation	16	0	0	0	0	0	0	0	16	143	0	*	*	0
hum-tgt-effect	124	386	35	20	11	1	18	320	58	26	36	12	83	20
hum-tgt-total-num	1	33	0	0	0	0	0	33	1	121	0	0	100	
inc-total	530	428	156	63	34	23	33	175	277	252	35	44	41	
perp-total	258	292	56	7	49	3	7	180	146	202	23	20	62	
phys-tgt-total	240	497	42	19	28	0	17	408	151	574	21	10	82	
hum-tgt-total	622	1574	156	97	85	5	86	1236	284	439	33	13	78	
MATCHED/MISSING	1650	1818	410	186	196	31	143	1026	858	1017	30	28	56	
MATCHED/SPURIOUS	919	2791	410	186	196	31	143	1999	127	936	55	18	72	
MATCHED ONLY	919	1818	410	186	196	31	143	1026	127	486	55	28	56	
ALL TEMPLATES	1650	2791	410	186	196	31	143	1999	858	1467	30	18	72	
SET FILLS ONLY	790	879	236	56	85	2	41	502	413	491	33	30	57	2
STRING FILLS ONLY	425	434	93	37	62	6	37	242	233	283	26	26	56	
TEXT FILTERING	69	99	68	*	*	*	*	31	1	0	98	69	31	100
							P&R		2P&R			P&2R		
F-MEASURES							22.5		19.57			26.47		

Table 1: Official TST3 score report.

The performance, on a slot by slot basis, is, therefore, what one might expect: the pure set fills such as INCIDENT: TYPE and INCIDENT: STAGE OF EXECUTION show much better performance than the string fills such as HUM TGT: NAME.

Table 2 shows the summary rows of the official template-by-template results on TST4. The complete official score report for TTS-MUC4 on TST4 can be found in Appendix G: Final Test Score Summaries. Performance was comparable on both sets of texts.

```
SLOT                     POS ACT|COR PAR INC|ICR IPA| SPU MIS  NON|REC PRE OVG FAL
-------------------------------+-----------+-------+------------+---------------
MATCHED/MISSING         1157 1260|340 146 157| 34  89|617 514 645  | 36  33  49
MATCHED/SPURIOUS         803 2273|340 146 157| 34  89|1630 160 955 | 51  18  72
MATCHED ONLY             803 1260|340 146 157| 34  89|617 160 404  | 51  33  49
ALL TEMPLATES           1157 2273|340 146 157| 34  89|1630 514 1196| 36  18  72
SET FILLS ONLY           561 612|195  48  77 |  0  31|292 241 314  | 39  36  48   2
STRING FILLS ONLY        302 293| 80  22  47 |  2  22|144 153 179  | 30  31  49
TEXT FILTERING            56  98| 56   *   * |  *   *| 42   0   2  |100  57  43  95
------------------------------+------------+-------+------------+---------------
                                                    P&R       2P&R       P&2R
F-MEASURES                                          24.0      20.0       30.0
```

Table 2: Summary rows of the official TST4 score report.

MUC-4 TEST SETTINGS

TTS-MUC4 uses Bayesian classifiers for each of the template slots. The general form for Bayesian classifiers is to compute,

$$\Pr\left(C_i | f_1 \wedge f_2 \ldots f_n\right)$$

where f_i are textual features. For set fill slots, the C_i are the possible values (*e.g.* DEATH, SOME DAMAGE, *etc.*). For the string fill slots, the C_i are yes or no answers to whether a particular item fills a slot, (*e.g.* HUMAN-TGT-NAME versus HUMAN-TGT-NAME-NOT). For typical Bayesian classifiers, the tunable parameter is the prior probabilities for the C_i. In TTS-MUC4 we have two different settings, EQUI-PROB and REL-FREQ, respectively for probabilities that are equal for all classes and probabilities that reflect the relative frequency of classes in the training data. EQUI-PROB favors recall, and REL-FREQ favors precision.

In addition, for text applications, there is an issue as to whether one includes only those features present in the text, or, also, those that are absent. In TTS-MUC4 we used two different settings, PRESENT and PRESENT&FREQUENT, where PRESENT&FREQUENT considers all those features which are present and also those that are absent, but which occur very frequently in the texts. The threshold for whether a feature was considered frequent was set so that, for each slot, approximately 30 features were considered frequent. In the TTS-MUC4 conceptual hierarchy there are over 400 potential features.

For each slot, the parameter settings were optimized to balance recall and precision. The optimization was done using TST1 and TST2. Table 3 gives the parameter settings for each slot. Balancing precision and recall for string fill slots is difficult in TTS-MUC4. For example, in the training corpus, TTS-MUC4 detects over 4,000 potential `HUMAN-TARGET-NAME`s, but less than 10% of these are actual string fills.

TRAINING METHODOLOGY

To compute the conditional probabilities, the MUC-3 development (DEV) corpus and the associated templates where used. Each sentence in the DEV corpus that contained a string fill for some template was used as a training sample. TTS detects features for important domain words (*e.g.* `explosion`, `report`, *etc.*), and also for phrases that *may* map into string fills. For each training sample, the presence or absence of each feature was examined to compute, for example,

$$\Pr_{rf}(\texttt{:explosion} - w|\texttt{:PHYS - TGT - TYPE =:COMMERCIAL})$$

The probability estimates using relative frequency, $\Pr_{rf}$, are then combined using Bayes rule on a new sentence to compute:

$$\Pr(C_i|f_1 \wedge f_2 \ldots f_n)$$

SLOT	Priors	Tests
INCIDENT-TYPE	REL-FREQ	PRESENT
STAGE-OF-EXEC	REL-FREQ	PRESENT
INSTRUMENT-ID	EQUI-PROB	PRESENT&FREQUENT
INSTRUMENT-TYPE	REL-FREQ	PRESENT&FREQUENT
PERP-INDIV	EQUI-PROB	PRESENT
PERP-ORG	EQUI-PROB	PRESENT
PERP-CAT	EQUI-PROB	PRESENT
PERP-CONF	EQUI-PROB	PRESENT&FREQUENT
HUM-TGT-NAME	EQUI-PROB	PRESENT
HUM-TGT-DESCR	EQUI-PROB	PRESENT
HUM-TGT-TYPE	REL-FREQ	PRESENT
HUM-TGT-EFFECT	REL-FREQ	PRESENT
PHYS-TGT-ID	EQUI-PROB	PRESENT&FREQUENT
PHYS-TGT-TYPE	REL-FREQ	PRESENT
PHYS-TGT-EFFECT	REL-FREQ	PRESENT

Table 3: Test run setting for the Bayesian classifiers.

In addition to training of the Bayesian classifiers, the DEV corpus was used, exactly as in TTS-MUC3, to derive phrase patterns for potential string fills. For example, "SIX JESUITS" would drive the creation of the phrase (:NUMBER-W :RELIGIOUS-ORDER-W). The type of the string fill served as the semantic feature for the phrase, which is :CIVILIAN-DESCR, in this example .

Improvement that occurred over time in TTS-MUC4 is attributable to two factors: the introduction of the Bayesian classifiers to replace the K-Neighbors technique from TTS-MUC3, and the tuning of the parameters of the Bayesian classifiers for each slot.

All of the training for TTS-MUC4 is automated. As with TTS-MUC3, the only manual portion of the process is choosing the conceptual classes for the lexicon.

ALLOCATION OF EFFORT

Two calendar months and approximately 2.5 person months were spent on enhancing the TTS-MUC3 system to create TTS-MUC4.

TTS-MUC4 effort falls roughly into three categories: classifier evaluation, system training, and filter development. Approximately 20% of our time was spent on developing and evaluating the performance of the Bayesian classifier, and tuning the parameters used in this classifier. This classifier replaced the K-Nearest Neighbor classifier previously employed in TTS-MUC3. 10% of the development effort focused on tuning other system parameters, such as the *fill-strength-threshold*, which provides a means for filtering out unlikely slot fillers. About 40% of our time was devoted to developing filters to improve the precision of the values of the template fillers, and evaluating their effects. Retraining of the system to take advantage of a modified lexicon and to accommodate the revised templates took up about 10% of the time. The remaining 20% of the effort was spent on developing code to extract information to fill the new and revised slots of the MUC-4 templates.

LIMITING FACTORS

One limiting factor for the Hughes TTS-MUC4 system was time. The Bayesian classifier is effective for filling most slots, but the K-Nearest Neighbor classifier might provide better fills for others. However, time did not permit us to experiment enough to identify the best classifier to use for each slot. Another aspect of TTS to which we would like to have devoted more attention is on dynamically weighting features retrieved from the knowledge base depending upon their relevance to the slot being processed. Our algorithm for grouping sentences into topics was responsible for many of our errors. Improving the slot-dependent weighting portion of the system would take a considerable amount of additional time, and would require that domain knowledge be added into the processing.

FUTURE WORK

The following enhancements are most relevant to the current MUC-oriented software: (1) filters for string fills based on linguistic knowledge, (2) reference resolution, and (3) better learning/pattern classification algorithms. TTS-MUC4 currently has a very limited amount of processing that is specialized for language. One of the features that we would have liked to detect in the MUC-4 corpus was the source of information in a story. Individuals who are the source of a report occurred frequently, and erroneously, as human targets. Another "language specific" portion we would like to add is reference resolution for string fills. TTS-MUC4 currently suffers in its precision score because it lists each referent for a filler several times.

Additional changes would make a more usable "real system", although they are not essential for the MUC task as it now stands. These include (1) the development of a user interface for corpus marking, and (2) integration with on-line data sources, such as map databases, to eliminate the burden of creating special data files for natural language processing.

TRANSFERABILITY TO OTHER TASKS

Currently, TTS only requires a lexicon and a training corpus with templates. Therefore, extension to terrorism in another locale or to a completely different domain would be easy. However, once features are added to improve performance, as noted in Section 6 above, handling a new domain will be more difficult.

LESSONS LEARNED

TTS-MUC4 represents a small increase in performance beyond TTS-MUC3. TTS currently has very little processing specific to language; most of the processing is simple feature detection followed, by pattern recognition algorithms. We believe that TTS-MUC4 represents a plateau in performance that will require more linguistic knowledge to increase performance. The goal for TTS, then, is to significantly increase performance without increasing development time for new applications.

REFERENCES

[1] Dolan, Charles P., Goldman, Seth R., Cuda, Thomas V., Nakamura, Alan M. Hughes Trainable Text Skimmer: description of the TTS system as used for MUC-3. *Proceedings of the Third Message Understanding Conference (MUC-3).* San Diego, California, 21-23 May 1991.

[2] Dolan, Charles P., Goldman, Seth R., Cuda, Thomas V., Nakamura, Alan M. Hughes Trainable Text Skimmer: MUC-3 test results and analysis. *Proceedings of the Third Message Understanding Conference (MUC-3).* San Diego, California, 21-23 May 1991.