UNIVERSITY OF MASSACHUSETTS: DESCRIPTION OF THE CIRCUS SYSTEM AS USED FOR MUC-3

Wendy Lehnert, Claire Cardie, David Fisher, Ellen Riloff, Robert Williams

Department of Computer and Information Science
University of Massachusetts
Amherst, MA 01003
lehnert@cs.umass.edu

BACKGROUND AND MOTIVATION

In 1988 Professor Wendy Lehnert completed the initial implementation of a semantically-oriented sentence analyzer named CIRCUS [1]. The original design for CIRCUS was motivated by two basic research interests: (1) we wanted to increase the level of syntactic sophistication associated with semantically-oriented parsers, and (2) we wanted to integrate traditional symbolic techniques in natural language processing with connectionist techniques in an effort to exploit the complementary strengths of these two computational paradigms.

Shortly thereafter, two graduate students, Claire Cardie and Ellen Riloff, began to experiment with CIRCUS as a mechanism for analyzing citation sentences in the scientific literature [2]. The key idea behind this work was to extract a relatively abstract level of information from each sentence, using only a limited vocabulary that was hand-crafted to handle a restricted set of target concepts. We called this mode of language processing *selective concept extraction,* and the basic style of sentence analysis was a type of text skimming. This project provided us with an opportunity to give CIRCUS a workout and determine whether or not the basic design was working as expected.

Although CIRCUS was subject to a number of limitations, the integration of syntax and semantics appeared to work very nicely. We believed we had constructed a robust text skimmer that was semantically oriented but nevertheless able to use syntactic knowledge as needed. Projects associated with the connectionist aspect of CIRCUS took off at about this time and carried us in those directions for a while [3,4,5]. When an announcement for MUC-3 reached us in June of 1990, we felt that the MUC-3 evaluation required selective concept extraction capabilities of just the sort we had been developing. We were eager to put CIRCUS to the test.

It was clear to us that MUC-3 would require a much more ambitious and demanding application of CIRCUS than our earlier work on citation sentences, and we fully expected to learn a great deal from the experience. We hoped to capitalize on Cardie and Riloff's previous experience with CIRCUS while identifying some new areas for ongoing research in sophisticated text analysis. In September of 1990, Robert Williams joined our MUC-3 effort as a post doc with research experience in case-based reasoning. Cardie, Riloff, and Williams provided the technical muscle for all of our MUC-3 system development and knowledge engineering. Cardie was primarily responsible for CIRCUS and dictionary design, Riloff developed the rule-based consolidation component, and Williams designed the case-based reasoning consolidation component. Although the division of labor was fairly clean, everyone worked with CIRCUS dictionary definitions and the preprocessor at various times as needed. In January of 1991, David Fisher joined the project as an undergraduate assistant who designed an interface for faster dictionary development while assisting with internal testing. Professor Lehnert assumed a leadership role but made no programming contributions to MUC-3.

SYSTEM COMPONENTS

Although CIRCUS was the primary workhorse underlying our MUC-3 effort, it was necessary to augment CIRCUS with a separate component that would receive CIRCUS output and massage that output into the final target template instantiations required for MUC-3. This phase of our processing came to be known as consolidation, although it corresponds more generally to what many people would call discourse analysis. We will describe both CIRCUS and our consolidation processing with examples from TST1-MUC3-0099. (Please consult Appendix H for the complete text of TST1-MUC3-0099.) A flow chart of our complete system is given in Figure 1.

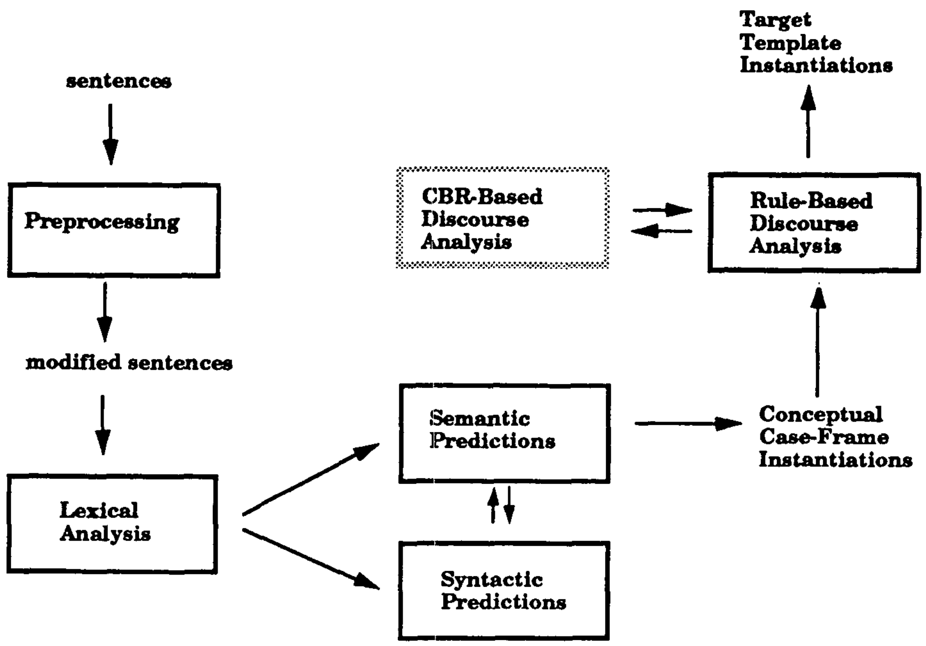
sentences
Preprocessing
modified sentences
Lexical Analysis
Semantic Predictions
Syntactic Predictions
CBR-Based Discourse Analysis
Rule-Based Discourse Analysis
Conceptual Case-Frame Instantiations
Target Template Instantiations

Figure 1: Flow Chart for the MUC-3/CIRCUS System

Sentence Preprocessing

To begin, each sentence is given to our preprocessor where a number of domain-specific modifications are made. (1) Dates are analyzed and translated into a canonical form. (2) Words associated with our phrasal lexicon are connected via underscoring. (3) Punctuation marks are translated into atoms more agreeable to LISP. For example, the first sentence (S1) reads:

S1: POLICE HAVE REPORTED THAT TERRORISTS TONIGHT BOMBED THE EMBASSIES OF THE PRC AND THE SOVIET UNION.

After preprocessing, we have:

S1: (POLICE HAVE REPORTED THAT TERRORISTS ON OCT_25_89 >CO TONIGHT BOMBED THE EMBASSIES OF THE PRC AND THE SOVIET_UNION >PE)

The canonical date was derived from "tonight" and the dateline of the article, "25 OCT 89." Most of our phrasal lexicon is devoted to proper names describing locations and terrorist organizations (831 entries). 25 additional proper names are also recognized, but not from the phrasal lexicon.

Lexical Analysis

At this point we are ready to hand the sentence to CIRCUS for lexical processing. This is where we search our dictionary and apply morphological analysis in an effort to recognize words in the sentence. Any words that are not recognized receive a default tag reserved for proper nouns in case we need to make sense out of unknown words later. In order for any semantic analysis to take place, we need to recognize a word that operates as a trigger for a concept node definition. If a sentence contains no concept node triggers, it is ignored by the semantic component. This is one way that irrelevant texts can be identified: texts that trigger no concept nodes are deemed irrelevant. Words in our dictionary are associated with a syntactic part of speech, a position or positions within a semantic feature hierarchy, possible concept node definitions if the item operates as a concept node trigger, and syntactic complement predictions. Concept nodes and syntactic complement patterns will be described in the next section. An example of a dictionary entry with all four entry types is our definition for "dead" as seen in figure 2.

```
(D-WORD DEAD
    :SYNTACTIC-TYPE SPECIAL-ADJECTIVE
    :SYNTACTIC-EXPECTATIONS
        (((assign *np-flag* t
                  *predicates* (append *predicates* (list *word*))
                  *part-of-speech* 'adjective
                  *cd-form* (make-special *word*)
                  *global-cn* *cd-form*)
          (next-packet
            ((test (eq *part-of-speech* 'noun)))
            ((test (eq *part-of-speech* 'adjective))))))
    :WORD-SENSES (dead1)
    :CN-DEFS ($LEFT-DEAD$ $FOUND-DEAD$ $FOUND-DEAD-PASS$))
```

Figure 2: The Dictionary Definition for "DEAD"

When morphological routines are used to strip an inflected or conjugated form back to its root, the root-form dictionary definition is dynamically modified to reflect the morphological information. For example, the root definition for "bomb" will pick up a :VERB-FORM slot with PAST filling it when the lexical item "bombed" is encountered.

Semantic and Syntactic Predictions

Words associated with concept nodes activate both syntactic and semantic predictions. In S1 the verb "bombed" activates semantic predictions in the form of a concept node designed to describe a bombing. Each concept node describes a semantic case frame with variable slots that expect to be filled by specific syntactic constituents. The concept node definition activated by "bombed" in S1 is given in Figure 3.

We can see from this definition that a case frame with variable slots for the actor and target is predicted. The actor slot expects to be filled by an organization, the name of a recognized terrorist or generic terrorist referent, a proper name, or any reference to a person. The target slot expects to be filled by a physical target. We also expect to locate the actor in the subject of the sentence, and the target should appear as either a direct object or the object of a prepositional phrase containing the preposition "in." None of these predictions will be activated unless the current sentence is in the active voice.

```
;; X bombed/dynamited/blew_up
;; the bomb blew up in the building (tst1-0040) -emr
;; (if this causes trouble we can create a new cn for blew_up)
(define-word $BOMBING-3$
    (CONCEPT-NODE
    ':NAME '$BOMBING-3$
    ':TIME-LIMIT 10
    ':SLOT-CONSTRAINTS '(((class organization *S*)(class terrorist *S*)
                            (class proper-name *S*)(class human *S*))
                          ((class phys-target *DO*) (class phys-target *PP*)))
    ':VARIABLE-SLOTS '(
                        ACTOR (*S* 1)
                        TARGET (*DO* 1 *PP* (is-prep? '(in))))
    ':CONSTANT-SLOTS '(
                        TYPE BOMBING))
    ':ENABLED-BY '((active))))
```

Figure 3: The $BOMBING-3$ Concept Node Definition

Syntactic complement predictions are managed by a separate mechanism that operates independently of the concept nodes. The syntactic predictions fill syntactic constituent buffers with appropriate sentence fragments that can be used to instantiate various concept node case frames. Syntactic predictions are organized in decision trees using test-action pairs under a stack-based control structure [6]. Although syntactic complements are commonly associated with verbs (verb complements), we have found that nouns should be used to trigger syntactic complement predictions with equal frequency. Indeed, any part of speech can trigger a concept node and associated complement predictions as needed. As we saw in the previous section, the adjective "dead" is associated with syntactic complement predictions to facilitate noun phrase analysis. Figure 4 shows the syntactic complement pattern predicted by "bombed" once morphology recognizes the root verb "bomb."

```
(((test (second-verb-or-infinitive?))   ;;  all verbs call this function
  (assign *part-of-speech* 'verb         ;;  just to be sure...reset some buffers for noun phrase collection

     *np-flag* nil *noun-group* nil  *predicates* nil *entire-noun-group* nil
     *determiners* nil *appositive* nil *gerund* nil

                                         ;;  verb assignments.

     *cd-form* (make-verb *word*)        *V* *cd-form*      *DO* nil)

 (next-packet                            ;;  next noun phrase should be the direct object

   ((test (equal *part-of-speech* 'noun-phrase))
    (assign *DO* *cd-form*))             ;;  don't predict *DO* if conjunction follows the verb,
                                         ;;  e.g., in "X was damaged and Y was destroyed",
                                         ;;  Y should NOT be *DO* of "damaged"
   ((test (equal *part-of-speech* 'conjunction))))))
```

Figure 4: The Verb Complement Pattern for "BOMBED"

Remarkably, Figure 4 displays all the syntactic knowledge CIRCUS needs to know about verbs. Every verb in our dictionary references this same prediction pattern. In particular, this means that we have found no need to distinguish transitive verbs from intransitive verbs, since this one piece of code handles both (if the prediction for a direct object fails, the *DO* buffer remains empty).

Once semantic and syntactic predictions have interacted to produce a set of case frame slot fillers, we then create a frame instantiation which CIRCUS outputs in response to the input sentence. In general, CIRCUS can produce an arbitrary number of case frame instantiations for a single sentence. No effort is made to integrate these into a larger structure. The concept node instantiation created by $BOMBING-3$ in response to S1 is given in Figure 5.

Some case frame slots are not predicted by the concept node definition but are inserted into the frame in a bottom-up fashion. Slots describing time specifications and locations are all filled by a mechanism for bottom-up slot insertion (e.g. the REL-LINK slot in Figure 5 was created in this way). Although the listing in Figure 5 shows only the head noun "embassies" in the target noun group slot, the full phrase "embassies of the PRC and the Soviet Union" has been recognized as a noun phrase and can be recovered from this case frame instantiation. The target value "ws-diplomat-office-or-residence" is a semantic feature retrieved from our dictionary definition for "embassy." No additional output is produced by CIRCUS in response to S1.

```
S1: (POLICE HAVE REPORTED THAT TERRORISTS ON OCT_25_89 >CO TONIGHT BOMBED THE
    EMBASSIES OF THE PRC AND THE SOVIET_UNION >PE)

      ***     TYPE = BOMBING
      ***     ACTOR = WS-TERRORIST
      ***     noun group = (TERRORISTS)
      ***     TARGET = WS-DIPLOMAT-OFFICE-OR-RESIDENCE
      ***     noun group = (EMBASSIES)
      ***     determiners = (THE)
      ***     REL-LINK (TIME  (OCT_25_89)))
```

Figure 5: CIRCUS Output for S1

It is important to understand that CIRCUS uses no sentence grammar, and does not produce a full syntactic analysis for any sentences processed. Syntactic constitutents are utilized only when a concept node definition asks for them. Our method of syntactic analysis operates locally, and syntactic predictions are indexed by lexical items. We believe that this approach to syntax is highly advantageous when dictionary coverage is sparse and large sentence fragments can be ignored without adverse consequences. This allows us to minimize our dictionaries as well as the amount of processing needed to handle selective concept extraction from open-ended texts.

Some concept nodes are very simple and may contain no variable slots at all. For example, CIRCUS generates two simple frames in response to S2, neither of which contain variable slot fillers.

Note that the output generated by CIRCUS for S2 as shown in Figure 6 is incomplete. There should be a representation for the damage. This omission is the only CIRCUS failure for TST1-MUC3-0099, and it results from a noun/verb disambiguation failure. The 14 sentences in TST1-MUC3-0099 resulted in a total of 27 concept node instantiations describing bombings, weapons, injuries, attacks, destruction, perpetrators, murders, arson, and new event markers.

```
S2: (THE BOMBS CAUSED DAMAGE BUT NO INJURIES >PE)

***     TYPE = WEAPON
***     INSTR = BOMB          (triggered by the noun "bombs")

***     TYPE = INJURY
***     MODE = NEG            (triggered by the noun "injuries")
```

Figure 6: CIRCUS Output for S2

Other Problems in Sentence Analysis

Special mechanisms are devoted to handling specific syntactic constructs, including appositives and conjunctions. We will illustrate our handling of conjunctions by examining two instances of "and" in S5:

S5: Police said the attacks were carried out almost simultaneously **and(1)** that the bombs broke windows **and(2)** destroyed the two vehicles.

We recognize that **and(1)** is not part of a noun phrase conjunction, but do nothing else with it. A new control kernel begins after "that" and reinitializes the state of the parser. **and2** is initially recognized as potentially joining two noun phrases --- "windows" and whatever noun phrase follows. However, when the verb "destroyed" appears before any conjoining noun phrase is recognized, the LICK mechanism determines that the conjunction actually joins two verbs and begins a new clause. As a result, the subject of "broke" (i.e., "the bombs") correctly becomes the subject of "destroyed" as well.

Rule-Based Consolidation

When an entire text has been processed by CIRCUS, the list of the resulting case frame instantiations is passed to consolidation. A rule base of consolidation heuristics then attempts to merge associated case frames and create target template instantiations that are consistent with MUC-3 encoding guidelines. It is possible for CIRCUS output to be thrown away at this point if consolidation does not see enough information to justify a target template instantiation. If consolidation is not satisfied that the output produced by CIRCUS describes bonafide terrorist incidents, consolidation can declare the text irrelevant. A great deal of domain knowledge is needed by consolidation in order to make these determinations. For example, semantic features associated with entities such as perpetrators, targets, and dates are checked to see which events are consistent with encoding guidelines. In this way, consolidation operates as a strong filter for output from CIRCUS, allowing us to concisely implement encoding guidelines independently of our dictionary definitions.

A number of discourse-level decisions are made during consolidation, including pronoun resolution and reference resolution. Some references are resolved by frame merging rules. For example, CIRCUS output from S1, S2 and S3 is merged during consolidation to produce the target template instantiation found in Figure 7.

The CIRCUS output from S1 triggers a rule called create-bombing which generates a template instantiation that eventually becomes the one in Figure 7. But to arrive at the final template, we must first execute three more consolidation rules that combine the preliminary template with output from S2 and S3. Pseudo-code for two of these three rules is given in Figure 8.

0. MESSAGE ID	DEV-MUC3-0099
1. TEMPLATE ID	1
2. DATE OF INCIDENT	25 OCT 89
3. TYPE OF INCIDENT	BOMBING
4. CATEGORY OF INCIDENT	TERRORIST ACT
5. PERPETRATOR: ID OF INDIV(S)	"TERRORISTS"
6. PERPETRATOR: ID OF ORG(S)	-
7. PERPETRATOR: CONFIDENCE	-
8. PHYSICAL TARGET: ID(S)	"EMBASSIES OF THE PRC AND THE SOVIETUNION" "PRC EMBASSY"
9. PHYSICAL TARGET: TOTAL NUM	PLURAL
10. PHYSICAL TARGET: TYPE(S)	DIPLOMAT OFFICE OR RESIDENCE: "EMBASSIES OF THE PRC AND THE SOVIET UNION" DIPLOMAT OFFICE OR RESIDENCE: "PRC EMBASSY"
11. HUMAN TARGET: ID(S)	-
12. HUMAN TARGET: TOTAL NUM	-
13. HUMAN TARGET: TYPE(S)	-
14. TARGET: FOREIGN NATION(S)	PEOPLES REP OF CHINA: "EMBASSIES OF THE PRC AND THE SOVIET UNION" PEOPLES REP OF CHINA: "PRC EMBASSY"
15. INSTRUMENT: TYPE(S)	*
16. LOCATION OF INCIDENT	PERU: LIMA (CITY): SAN ISIDRO (NEIGHBORHOOD)
17. EFFECT ON PHYSICAL TARGET(S)	-
18. EFFECT ON HUMAN TARGET(S)	NO INJURY OR DEATH: "-"

Figure 7: Our Output Template Representation for S1-S3

```
==========================================================================
MERGE-WEAPON-BOMBING

IF  $weapon structure
 and the weapon is an explosive
 and a BOMBING or ATTEMPTED-BOMBING template  is on the stack in the current family
 and dates are compatible and locations are compatible

THEN
  merge instruments, dates, and locations

==========================================================================
MERGE-BOMBING-BOMBING

IF  $bombing structure
 and BOMBING template  is on the stack in the current family
 and dates are compatible and locations are compatible

THEN
  merge perpetrators, human targets, physical targets, instruments, dates, and locations
 and also ...
  if there is a MURDER template with compatible victim (on the stack in the same family)
          with no instruments or the instruments are explosives
  then
  merge perpetrators, human targets, instruments, dates, and locations with the MURDER
==========================================================================
```

Figure 8: Two Merging Rules from Rule-Based Consolidation

Note also that the location of the incident was merged into this frame from S3 which triggers another bombing node in response to the verb "exploded" as shown in figure 9.

S3: (A CAR_BOMB EXPLODED IN_FRONT_OF THE PRC EMBASSY >CO IN THE LIMA RESIDENTIAL DISTRICT OF SAN_ISIDRO >PE)

```
***     TYPE = BOMBING
***     INSTR =
 >>>        TYPE = WEAPON
 >>>        INSTR = CAR_BOMB
 >>>        REL-LINK (TARGET OBJECT (WS-DIPLOMAT-OFFICE-OR-RESIDENCE))
 >>>        REL-LINK (LOC2 OBJECT (WS-GENERIC-LOC))
***     noun group = (CAR_BOMB)
***     determiners = (A)
***     REL-LINK (TARGET  (PRC EMBASSY)))
***     REL-LINK (LOC2  (LIMA DISTRICT)))
```

Figure 9: CIRCUS Output for S3

Once again, the top-level REL-LINK for a location is printing out only a portion of the complete noun phrase that was captured.

Although we would say that the referent to "bombs" in S2 is effectively resolved during consolidation, our methods are not of the type normally associated with linguistic discourse analysis. When consolidation examines these case frames, we are manipulating information on a conceptual rather than linguistic level. We need to know when two case frame descriptions are providing information about the same event, but we aren't worried about referents for specific noun phrases per se.

We did reasonably well on this story. Three templates of the correct event types were generated and no spurious templates were created by the rule base. Sentences S9 through S13 might have generated spurious templates if we didn't pay attention to the dates and victims. Here is how the preprocessor handled S12:

S12: IN ANOTHER INCIDENT 3 YEARS AGO, A SHINING PATH MILITANT WAS KILLED BY SOVIET EMBASSY GUARDS INSIDE THE EMBASSY COMPOUND.

S12: (IN ANOTHER INCIDENT ON -DEC_31_80 >CO &&3 YEARS AGO >CO >CO A SHINING_PATH MILITANT WAS KILLED BY SOVIET EMBASSY GUARDS INSIDE THE EMBASSY COMPOUND >PE)

Whenever the preprocessor recognizes a date specification that is "out of bounds" (at least two months prior to the dateline), it inserts -DEC_31_80 as a flag to indicate that the events associated with this date are irrelevant. This date specification will then be picked up by any concept node instantiations that are triggered "close" to the date description. In this case, the event is irrelevant both because of the date and because of the the victim (murdered militants aren't usually relevant). Despite the fact that S10 and S13 contain no date descriptions, the case frames generated for these sentences are merged with other frames that do carry disqualifying dates, and are therefore handled at a higher level of consolidation. In the end, the two murders (S11 and S12) are discarded because of disqualifications on their victim slot fillers, while the bombing (S9) was discarded because of the date specification. The injuries described by S10 are correctly merged with output from S9, and therefore discarded because of the date disqualifier. Likewise, the dynamite from S13 is correctly merged with the murder of the militant, and the dynamite is subsequently discarded along with the rest of that template.

Case-Based Consolidation

The CBR component of consolidation is an optional part of our system, designed to increase recall rates by generating additional templates to augment the output of rule-based consolidation. These extra templates are generated on the basis of correlations between CIRCUS output for a given text, and the key target templates for similarly indexed texts. The CBR component uses a case base which draws from 283 texts in the development corpus, and the 100 texts from TST1 for a total of 383 texts. We experimented with a larger case base but found no improvement in performance. The case base contains 254 template type patterns based on CIRCUS output for the 383 texts in the case base.

Each case in the case base associates a set of concept nodes with a template containing slot fillers from those concept nodes. The concept nodes are generated by CIRCUS when it analyzes the original source text. A case has two parts: (1) an incident type, and (2) a set of sentence/slot name patterns. For example, suppose a story describes a bombing such that the perpetrator and the target were mentioned in one sentence, and the target was mentioned again three sentences later. The resulting case would be generated in response to this text:

```
BOMBING
    0: (PERP TARGET)
    3: (TARGET)
```

The numerical indices are relative sentence positions. The same pattern could apply no matter where the two sentences occurred in the text, as long as they were three sentences apart.

Cases are used to determine when a set of concept nodes all contribute to the same output template. When a new text is analyzed, a probe is used to retrieve cases from the case base. Retrieval probes are new sentence/slot name patterns extracted from the current CIRCUS output. If the sentence/slot name pattern of a probe matches the sentence/slot name pattern of a case in the case base, that case is retrieved, the probe has succeeded, and no further cases are considered.

Maximal probes are constructed by grouping CIRCUS output into maximal clusters that yield successful probes. In this way, we attempt to identify large groups of consecutive concept nodes that all contribute to the same output template. Once a maximal probe has been identified, the incident type of the retrieved case forms the basis for a new CBR-generated output template whose slots are filled by concept node slot fillers according to appropriate mappings between concept nodes and output templates.

In the case of TST1-MUC3-0099, case-based consolidation proposes hypothetical templates corresponding to 3 bombings, 2 murders, 1 attack, and 1 arson incident. Two of the bombings and the arson are discarded because they were already generated by rule-based consolidation. The two murders are discarded because of victim and target constraints, while the third bombing is discarded because of a date constraint. The only surviving template is the attack incident, which turns out to be spurious. It is interesting to note that for this text, the CBR component regenerates each of the templates created by rule-based condolidation, and then discards them for the same reasons they were discarded earlier, or because they were recognized to be redundant against the rule-based output. We have not run any experiments to see how consistently the CBR component duplicates the efforts of rule-based consolidation. While such a study would be very interesting, we should note that the CBR templates are generally more limited in the number of slot fillers present, and would therefore be hard pressed to duplicate the overall performance of rule-based consolidation.

CONCLUSIONS

As we explained at the beginning of this paper, CIRCUS was originally designed to investigate the integration of connectionist and symbolic techniques for natural language processing. The original connectionist mechanisms in CIRCUS operated to manage bottom-up slot insertion for information found in unexpected (i.e. unpredicted) prepositional phrases. Yet when our task orientation is selective concept extraction, the information we are trying to isolate is strongly predicted, and therefore unlikely to surface in a bottom-up fashion. For MUC-3, we discovered that bottom-up slot insertion was needed primarily to handle only dates and locations: virtually all other relevant information was managed in a predictive fashion. Because dates and locations are relatively easy to recognize, any number of techniques could be successfully employed to handle bottom-up slot insertion for MUC-3. Although we used the numeric relaxation technique described in [1] to handle dates and locations, we consider this mechanism to be excessively powerful for the task at hand, and it could readily be eliminated for efficiency reasons in a practical implementation.

Although our score reports for TST2 indicate that our system is operating at the leading edge of overall performance for all MUC-3 systems, we nevertheless acknowledge that there are difficulties with our approach in terms of system development. It would take us a lot of hard work (again) to scale up to this same level of performance in a completely new domain. New and inexperienced technical personnel would probably require about 6 months of training before they would be prepared to attempt a technology transfer to a new domain. At that point we would estimate that another 1.5 person/years of effort would be needed to duplicate our current levels of performance in a new domain. Although these investments are not prohibitive, we believe there is room for improvement in the ways that we are engineering our dictionary entries and rule-based consolidation components. We need to investigate strategies for deducing linguistic regularities from texts and explore available resources that might leverage our syntactic analysis. Similar steps should be taken with respect to semantic analysis although we are much more skeptical about the prospects for sharable resources in this problem area.

Although we have had very little time to experiment with the CBR consolidation component, the CBR approach is very exciting in terms of system development possibilities. While the rule-based consolidation component had to be crafted and adjusted by hand, the case base for the CBR component was generated automatically and required virtually no knowledge of the domain or CIRCUS per se. In fact, our CBR module can be transported with minor modification to any other MUC-3 system that generates case frame meaning representations for sentences. As a discourse analysis component, this module is truly generic and could be moved into a new domain with simple adjustments. The labor needed to make the CBR component operational is the labor needed to create a development corpus of texts with associated target template encodings (assuming a working sentence analyzer is already in place). It is much easier to train people to generate target templates for texts than it is to train computer programmers in the foundations of artificial intelligence and the design of large rule bases. And the amount of time needed to generate a corpus from scratch is only a fraction of the time needed to scale up a complicated rule base. So the advantages of CBR components for discourse analysis are enticing to say the least. But much work needs to be done before we can determine the functionality of this technology as a strategy for natural language processing.

Having survived the MUC-3 experience, we can say that we have learned a lot about CIRCUS, the complexity of discourse analysis, and the viability of selective concept extraction as a technique for sophisticated text analysis. We are encouraged by our success, and we are now optimally positioned to explore exciting new research areas. Although our particpation in MUC-3 has been a thoroughly positive experience, we recognize the need to balance intensive development efforts of this type against the somewhat riskier explorations of basic research. We would not expect to benefit so dramatically from another intensive performance evaluation if we couldn't take some time to first digest the lessons we have learned from MUC-3. Performance evaluations can operate as an effective stimulus for research, but only if they are allowed to accompany rather than dominate our principal research activities.

ACKNOWLEDGEMENTS

Our participation in MUC-3 was supported by a DARPA contract administered by Meridian Aggregates Co., Contract No. MDA903-89-C-0041, the Office of Naval Research, under a University Research Initiative Grant, Contract No. N00014-86-K-0764 and NSF Presidential Young Investigators Award NSFIST-8351863.

BIBLIOGRAPHY

[1] Lehnert, W.G., "Symbolic/Subsymbolic Sentence Analysis: Exploiting the Best of Two Worlds," Technical Report No. 88-99, Department of Computer and Information Science, University of Massachusetts. 1988. Also available in *Advances in Connectionist and Neural Computation Theory,* Vol. I. (ed: J. Pollack and J. Barnden). pp. 135-164. Ablex Publishing, Norwood, New Jersey. 1991.

[2] Lehnert, W., Cardie, C. and Riloff, E., "Analyzing Research Papers Using Citation Sentences," in *Proceedings of the Twelfth Annual Conference of the Cognitive Science Society,* Boston MA. pp. 511-518. 1990.

[3] Wermter, S., "Integration of Semantic and Syntactic Constraints for Structural Noun Phrase Disambiguation", in *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence,* pp. 1486-1491. 1989.

[4] S. Wermter, "Learning Semantic Relationships in Compound Nouns with Connectionist Networks", in *Proceedings of Eleventh Annual Conference on Cognitive Science* , pp. 964-971. 1989.

[5] Wermter, S. and Lehnert, W.G., "A Hybrid Symbolic/Connectionist Model for Noun Phrase Understanding," in *Connection Science,* Vol. 1, No. 3. pp. 255-272. 1989.

[6] Riesbeck, C., "Micro ELI", in *Inside Computer Understanding,* (eds: R. Schank and C. Riesbeck) pp. 354-372. Lawrence Erlbaum Associates, 1981.