# DESCRIPTION OF THE UMASS SYSTEM AS USED FOR MUC-6

*David Fisher, Stephen Soderland, Joseph McCarthy, Fangfang Feng and Wendy Lehnert*

Box 34610, Department of Computer Science
University of Massachusetts, Amherst MA 01003
dfisher, soderlan, jmccarthy, feng, lehnert@cs.umass.edu
413-545-3639

## INTRODUCTION

Information extraction research at the University of Massachusetts is based on portable, trainable language processing components. Some components are more effective than others, some have been under development longer than others, but in all cases, we are working to eliminate manual knowledge engineering. Although UMass has participated in previous MUC evaluations, all of our information extraction software has been redesigned and rewritten since MUC-5, so we are evaluating a completely new system this year.

In particular, we are working with new string recognition specialists (for Named Entities), a new part-of-speech tagger, a new sentence analyzer, a new and fully automated dictionary construction algorithm, a new discourse analyzer, and a new coreference analyzer. The most interesting components in our system are CRYSTAL (which generates a concept node dictionary) [13], WRAP-UP (which establishes relational links between entities) [14, 15, 16], and RESOLVE (the coreference analyzer) [8]. Each of these components utilizes machine learning techniques in order to customize crucial extraction capabilities based on representative training texts.

Our preparations for MUC-6 began on June 19 (at the release of the Call for Participation) and ended on October 2 when we began our test runs. All of our ST-specific training began in September with the release of the ST keys. As much as we try to exploit trainable technologies, there are nevertheless places where some amount of manual coding is still needed. For example, we needed to write string manipulation functions to trim output strings in an effort to generate slot fills consistent with the MUC-6 slot fill guidelines. We also needed to create template parsers and text marking interfaces in order to map the MUC-6 training documents into data usable by our trainable components.

## SYSTEM OVERVIEW

At the foundation of all our system configurations are the string specialists: these are the pattern matching routines that attempt to recognize proper names, dates, and other stylized noun phrase descriptions. The specialists that we used for NE consist of separate routines designed to handle locations, organizations, people, dates, money, and percentages. The location, organization, and people specialists rely on dictionaries for their recognition. The organization and people specialists used for NE were based on code (heavily modified) developed in the Information Retrieval Laboratory at UMass. The dictionaries supporting those specialists were also borrowed from the IR Lab with some adjustments for MUC-6. All other specialists were developed in the NLP Lab, and the location specialist accessed a dictionary based on a subset of the Gazetteer entries.

The BADGER sentence analyzer refers to a collection of processes associated with part-of-speech (p-o-s) tagging, a trainable decision tree used to locate appositive constructions, local syntactic analysis, and semantic case frame instantiation. The processing of BADGER is not significantly different than that of the CIRCUS system used in previous MUC evaluations [4, 5, 6]. Concept node (CN) definitions are still used to create case frame instantiations and multiple CN definitions can apply to the same text fragment. BADGER is domain/task independent and requires no adjustment in order to move from one application to another. It does depend on CN definitions that are appropriate for a given domain/task, but different CN dictionaries can be plugged into BADGER as fully portable dictionary files. BADGER also relies on a p-o-s dictionary as well as semantic feature tags, and we do customize a p-o-s dictionary and a semantic feature hierarchy for specific domains. This customization is handled manually, but it is not a difficult task.

BADGER currently recognizes 27 p-o-s tags and it took 24 hours to manually create a p-o-s dictionary for MUC-6. We began with our domain independent core tags lexicon, which includes prepositions, determiners, and a number of common verbs. We then added terms that occurred 500 or more times in the six years of Wall Street Journal articles (1987-1992) from the Tipster collection, based on manual inspection. The p-o-s tag lexicon has 2084 entries. Semantic features were assigned to those same terms and also to those terms of interest to the ST task that appeared in the formal training set (100 texts). The semantic lexicon has 5453 entries. We used 45 semantic features and the creation of this semantic tagging dictionary took 36 hours.

The creation of a CN dictionary is now fully automated and accomplished by CRYSTAL, an inductive dictionary construction system. In previous MUC evaluations we used the AutoSlog system to generate CN dictionaries [10, 11, 12], but AutoSlog required human interaction for a quality control assessment of proposed CN definitions. CRYSTAL requires no such human review, and creates CN dictionaries on the basis of machine learning techniques [13].

In order to map BADGER output into MUC-6 text annotations, a number of decisions must be made about important noun phrases, including semantic type identification, concept attribute assignments, and coreference recognition. These decisions are handled primarily by hand-coded consolidation routines, WRAP-UP, a trainable discourse analyzer designed to establish relational links between referents, and RESOLVE, a trainable coreference analyzer.

When WRAP-UP makes a decision about the proper role of a given noun phrase or a possible relationship between two noun phrases, it considers evidence from a variety of sources. All of the features used by WRAP-UP are extracted using a domain-independent mechanism to encode features from CN slot values, from the relative position of the referents, and from verb patterns in which the noun phrase appeared. Sometimes these various sources provide consistent interpretations, but they often disagree with one another. All of WRAP-UP's decisions are handled by trained decision trees so discrepancies in the incoming data are managed on the basis of similar situations encountered during training. WRAP-UP is most effective when its training has given it the experience it needs to deal with various kinds of incongruencies.

RESOLVE handles all the crucial merging decisions used by consolidation and WRAP-UP. It determines when two noun phrases refer to the same entity and should therefore be merged in order to consolidate feature descriptors into a single entity description. If RESOLVE merges entities too aggressively, recall will fall, and if RESOLVE is too passive in its merging decisions, precision will suffer due to spurious entities. RESOLVE's decisions are based on a decision tree induced from feature vector representations of noun phrases. Some of the features used in these representations are domain independent, and some are created for specific domains. The version of RESOLVE used for TE and ST relied on a subset of its domain-independent feature set rather than the larger, domain-enhanced feature set used for the CO task. TE and ST would probably have benefited from the larger feature set used for the CO task, but there was not enough time to incorporate all of RESOLVE’s features into the systems used for the TE and ST tasks.

Space prohibits us from going into much detail about each of the major components in our various MUC-6 system configurations, so we will concentrate on the trainable components that were present in the TE, CO, and ST tasks. Since different tasks required different configurations, we summarize the relevant system components in the table below:

| | NE | CO | TE | ST |
|---|---|---|---|---|
| String specialists | x | x | x | x |
| BADGER (sentence analysis) | x | x | x | |
| RESOLVE (coreference) | | x | x | x |
| CRYSTAL (dictionary construction) | | x | x | |
| WRAP-UP (relational links) | | | x | |

We also note that some major system components were designed and implemented during our preparations for MUC-6, including all the string specialists, all the consolidation routines, template parsers and text-marking interfaces to translate MUC-6 training documents into formats compatible with our various trainable components, and a completely new implementation of WRAP-UP.

## A SYSTEM WALK-THROUGH

It always helps to look at concrete examples, and the designated walk-through text provides us with an opportunity to describe some selected processing as it tackles a specific text. We will limit most of our discussion to a single sentence as it was handled by RESOLVE, CRYSTAL, and WRAP-UP.

> "Mr. James, 57 years old, is stepping down as chief executive officer
> on July 1 and will retire as chairman at the end of the year."

## A Walk With RESOLVE

RESOLVE is needed here to determine that "Mr. James", "chief executive officer" and "chairman" are all descriptions of the same person entity. RESOLVE succeeds here by examining pairwise combinations of noun phrases (NPs).

The pairwise examination of NPs is handled by a C4.5 decision tree. This tree was created by an inductive algorithm in response to a collection of representative NP pairs extracted from available training data. The complete tree is large, containing 133 nodes or possible decisions points. To get a feel for the RESOLVE d-tree, we present a piece of the tree starting at the root node. The parenthetical numbers indicate how many training instances were encountered at each leaf nodes of the tree. Decision points are more reliable when a large number of training instances are examined for a given condition.

Tree 1: a portion of the RESOLVE coreference tree

```
ALIAS = YES:  "+"     (122.0/1.0)
ALIAS = NO:
|     SAME-STRING = YES:  "+"     (73.0/4.0)
|     SAME-STRING = NO:
|     |     MOST-RECENT-COMPATIBLE-SUBJECT = YES:
|     |     |     NAME-2 = NO:  "+"     (46.0/1.0)
|     |     |     NAME-2 = YES:
|     |     |     |     NAME-1 = NO:  "+"     (3.0/1.0)
|     |     |     |     NAME-1 = YES:  "-"     (3.0)
|     |     MOST-RECENT-COMPATIBLE-SUBJECT = NO:
|     |     |     SAME-TYPE = NO:  "-"     (1531.0)
|     |     |     SAME-TYPE = YES:
|     |     |     |     PERSON-IS-ROLE = YES:  "+"     (12.0)
|     |     |     |     PERSON-IS-ROLE = NO:
```

The ALIAS feature is the root node of the tree. This feature is true when an NP is a recognized alias of the other NP (e.g. "GM" is an alias of "General Motors Corp."). After checking ALIAS and SAME-STRING, the feature MOST-RECENT-COMPATIBLE-SUBJECT is checked. This feature is true when phrase-1 and phrase-2 are compatible (number and gender), and when phrase-1 is the most recent SUBJECT in the text. This feature was included specifically to handle pronoun resolution, and represents a variation on the well-known heuristic of merging with any referent found in the most recent compatible phrase. However, RESOLVE's version adds an extra constraint: the previous phrase must be found in the SUBJECT buffer.

If phrase-1 is the MOST-RECENT-COMPATIBLE-SUBJECT, and NAME-2 = No (i.e., phrase-2 has no name information - meaning that it's probably a pronoun or other anaphoric reference), then the phrases are judged coreferent. If phrase-2 **does** have name information (NAME-2 = Yes), but phrase-1 does **not** have name information (NAME-1 = No), then they are also judged coreferent.

It is interesting to note that SAME-TYPE is not checked until the fourth layer of the tree: if SAME-TYPE = No, then they are **not** coreferent. One might reasonably expect to see this at the root node since incompatible types should certainly be strong evidence for non coreference. But keep in mind that RESOLVE was not trained on perfect data. If NPs were not processed correctly by the string specialists, then RESOLVE is right to reduce its confidence in the SAME-TYPE feature accordingly. We know from our NE evaluation, that our string specialists sometimes labeled an organization as a person or location by mistake. This created a significant noise level for SAME-TYPE, enough to render the feature questionable. Had SAME-TYPE been more reliable, it probably would have found its way to the top of the tree.

Here is a rundown of how the features at the top of RESOLVE's decision tree were operating during the complete walkthrough text:

> ALIAS = YES was used for 25 instances. Every instance was correctly classified, though 4 instances were scored incorrect due to faulty string trimming.
>
> SAME-STRING = YES was used for 19 instances. 14 of those were correctly classified; all the misclassified instances were "it" phrases, so a little contextual knowledge would have probably helped (all "it" phrases were attempted, but many were irrelevant).
>
> MOST-RECENT-COMPATIBLE-SUBJECT = YES was used for 2 instances. Both times it misclassified the instances. This feature was supposed to apply only to pronouns and generic descriptions ("the company"), but in looking over the code for this feature extractor, we see that it was not properly constrained.
>
> PERSON-IS-ROLE = YES This feature was never used in the walk-through text. If the patterns used in this feature extractor were expanded (to include, for example, "IS STEPPING DOWN AS"), then it might have been more useful.

To understand how RESOLVE handled our focus sentence, we have to examine portions of the tree further from the root. It appears that the following decision points were important for this sentence:

Tree 2: another portion of the coreference tree

```
|   SAME-TYPE = YES:
|   |   PRONOUN-2 = NO:
|   |   | | | | PRONOUN-1 = NO:
|   |   | | | | |   NAME-2 = NO:
|   |   | | | | |   |   | | SAME-SENTENCE = YES:
|   |   | | | | |   |   | | |   NAME-1 = YES:  "+"   (20.0/8.0)
```

Some intermediate nodes have been removed (all had value "NO"). These branch points indicate that anytime we had two references to the same type of object, neither phrase is a pronoun, the second phrase is not a proper name, both are in the same sentence, and the first phrase is a proper name, then RESOLVE classified the two references as coreferent. The numbers at the leaf node (NAME-1 = YES) indicate that there were 20 instances that had the same feature values in the training set, and that 12 of these were positive and 8 were negative. Since the default classification for an ambiguous leaf node -- a leaf node that contained both positive and negative instances -- was to take the "majority class", the tree returned "+".

This may seem to be an unintuitive and risky pattern for coreference classification, but in fact, this processing turned out to be correct in the walk-through text whenever RESOLVE received correctly extracted NPs. The only time it erred was when RESOLVE was handed badly trimmed phrases or phrases with incorrect semantic features. So we've stumbled upon a rule induced by RESOLVE that probably wouldn't have been discovered manually. Yet it appears to be effective on real data -- which is precisely why machine learning may be more effective in the long run than manual knowledge engineering.

## A Walk with CRYSTAL

When we look at the full sentence analysis for our target sentence, we begin with a simple syntactic analysis into two segments:

| | |
|---|---|
| Subj: | Mr. James, 57 years old |
| Verb: | is |
| Obj: | stepping |
| PP: | down as chief executive officer |

| | |
|---|---|
| Conj: | and |
| Verb: | will retire |
| PP: | as chairman |
| PP: | at the end of the year |

Note that BADGER recognized “stepping” as a noun. This is because our p-o-s dictionary (derived from WSJ articles) didn’t contain “stepping” as a verb form. Interestingly, this error does not cause us any difficulties downstream because the exact same interpretation would have been applied during training. As long as the system is handling noun/verb ambiguities consistently, we do not suffer substantially from these tagging errors.

BADGER then applies CN definitions from CRYSTAL's dictionary and finds four CNs that apply to the first segment, three that extract "Mr. James, 57 years old" and one that extracts "chief executive officer".

```
Mr. James, 57 years old, is stepping down as chief executive officer on July 1
  PERSON-NAME                                 POSITION-NAME
  STATUS-OUT
  ON_THE_JOB-YES
```

These are the CN definitions that applied here:

```
Extract PERSON-NAME from Subj
    if Subj head class is Person_Name.

Extract STATUS-OUT from Subj [1]
    if Subj head class is Person_Name
       Verb contains IS
       Obj contains STEPPING

Extract ON_THE_JOB-YES from Subj [1]
    if Subj head class is Person_Name
       Verb contains IS
       Obj contains STEPPING

Extract POSITION-NAME from Prepositional Phrase if
       Prep is DOWN, head class is Person_Role
```

WRAP-UP can easily link the first three into an In_and_Out relationship, based solely on positional features, since they extracted identical buffers. WRAP-UP was also able to link the Position "chief executive officer" with this In_and_Out, based primarily on positional information.

BADGER also finds that two CNs apply to the second segment, one extracting "will retire" as STATUS_EVIDENCE-OUT and the other extracting "chairman" as POSITION-NAME. Since there is no explicit person name in this segment, no STATUS CNs or ON_THE_JOB CNs apply. WRAP-UP decision trees had difficulty determining which person to link with a Status_Evidence, and tended to attach the Status_Evidence to all persons in the same sentence.

The CRYSTAL dictionary that was trained on the 300 TE texts generated 2945 CN definitions. An additional 596 task-specific CN definitions were learned from the 100 ST training texts.

## A Walk with WRAP-UP

With these CNs in hand, WRAP-UP can then apply its trained d-trees to the CNs in order to establish relational links between objects. WRAP-UP used 17 different C4.5 decision trees in its processing. Eight of these identify specific relationships between entities, five of them attempt to filter out spurious entities that do not meet the scenario relevance guidelines, and four fill in default values for template element attributes. We will now look at two examples of selected trees in action.

Tree 3 shows a portion of a tree that considers relationships between Person and Status. Each Status or Status_Evidence which has been identified in the text is paired with each Person to form an instance. An instance is formed for "Mr. James" paired with the Status-Out from the "X is stepping down" concept node. This tree returns a negative classification if the Person and Status are not found in the same sentence. If both are found in the same noun phrase (which is the case for Mr. James and Status-Out), the tree returns a positive classification. An In-and-Out relationship is created with IO_Person = Mr. James and New_Status = Out.

If the Person was not from the same noun phrase as a Status CN, the tree returns negative. If the CN was of type Status_Evidence, as was the case for "will retired" in the second segment of the sentence, the tree branches to a large subtree in which two thirds of the training instances were positive. The tree returns a positive classification for the instance with "Mr. James" and "will retire". This leads to an In-and-Out relationship which is eventually merged with the textually identical In-and-Out already created.

Tree 3: a portion of the Person-Status-Links tree

```
sentences_apart  > 0 :  "-"
sentences_apart <= 0 :
|   |   |   same_noun_phrase = YES:  "+"
|   |   |   same_noun_phrase =  NO:
|   |   |   |   Obj2-CN_type-STATUS = YES:  "-"
|   |   |   |   Obj2-CN_type-STATUS =  NO:
```

Tree 4 is from the Filter stage of WRAP-UP and classifies Persons as relevant ("+") or irrelevant ("-"). The main criterion for relevance is whether a Person is involved in a management succession event, which is reflected in this tree. Persons with multiple links from In_and_Out objects are classified as relevant. By the time this tree is applied, "Mr. James" from our example has been linked to more than one In-and-Out and is classified as relevant. Each of persons from this text who are not involved in a change of status were correctly classified as irrelevant by WRAP-UP and were discarded.

Tree 4: a portion of the Person-Filter tree

```
Links-from-In_and_Out  > 1:  "+"
Links-from-In_and_Out <= 1:
|     Links-from-In_and_Out  <= 0:  "-"
|     Links-from-In_and_Out  > 0:
|     |     X-SAYS = YES:  "-"
|     |     X-SAYS =  NO: ...
```

## SCORE REPORTS AND DISCUSSION

We participated in all four MUC-6 evaluations in order to obtain as much feedback about our various components as possible. In an effort to reach beneath the numbers of the score reports, we conducted a few informal comparison-point experiments which we will report here as well.

## Named Entities (NE)

The NE task was handled by four independent string specialists designed and implemented during our MUC-6 preparations. Dates, money, and percentages were all handled by a single specialist. A breakdown of our recall and precision for each specialist is shown below:

| | R | P |
|---|---|---|
| organizations | 57 | 75 |
| people | 97 | 97 |
| locations | 82 | 76 |
| dates | 93 | 96 |
| money | 99 | 97 |
| percentages | 100 | 68 |
| total (all objects) | 82 | 89 |

Our string specialists were organized in a serial architecture which allowed upstream components to claim strings in a non-negotiable manner. The chain of specialists operated in the following order:

[money/dates/percentages] ---> organizations ---> people ---> locations

The numeric specialists were reliable and did not interfere with downstream components by claiming false hits. However, the low coverage of the organization specialist's dictionary left many organization names free to be claimed by the person and location specialists. This introduces an interference effect into the precision scores for each of the three ENAMEX types. Text claimed by the wrong specialist, say an organization name marked as a location, is counted as an incorrect organization type.

To see how each specialist was performing individually, we broke down the ACTUAL column of the score report into three columns, one for each of the three specialists. The resulting R/P scores are as follows, with P1 the precision for the type as reported by the scoring program, and P2 the precision for the individual specialists. P2 is computed as the number correct for a type divided by the objects reported as having that type.

| SLOT | POSSIBLE | ACTUAL | | | COR | REC | P1 | P2 |
|---|---|---|---|---|---|---|---|---|
| | | org | per | loc | | | | |
| organization | 439 | 260 | 40 | 31 | 249 | 57 | 75 | 93 |
| person | 373 | 0 | 372 | 3 | 363 | 97 | 97 | 87 |
| location | 109 | 7 | 3 | 109 | 90 | 82 | 76 | 63 |
| Total | | 267 | 415 | 143 | | | | |

From this table we see that the organization specialist actually made the fewest classification errors, misclassifying only seven locations. Names that slipped past the organization specialist and were claimed by the person or location specialists were charged against organization precision by the scoring program. So the organization specialist was penalized twice for those errors.

While it is apparent that the serial architecture is far from ideal, our most glaring weakness was the recall of the organization specialist. We had not anticipated this problem on the basis of the dry run materials. A post mortem of the official NE test set shows that it contained a large number of government organizations, which represented a weak spot in our organization dictionary. This was a regrettable oversight on our part, which undoubtedly hurt recall for CO, TE, and ST, given the importance of organization entities throughout.

We also note that the precision of the money and percentage specialists would have been perfect had we succeeded in filtering out a data table in one of the test texts. The specialists were not at fault for that filtering error which failed to follow stated extraction guidelines.

## Coreference (CO)

RESOLVE is a coreference resolution system that uses machine learning techniques to determine coreferent relationships among relevant phrases in a text. For the MUC-6 evaluation, we used the C4.5 decision tree induction system [9]. RESOLVE was designed to work in conjunction with an information extraction system; as such, its expected input is a set of phrases that are relevant to a specified information extraction task. The knowledge RESOLVE uses in order to learn to classify coreferent phrases is based on the same shallow knowledge used by our other system components.

The MUC-6 CO task was defined to include nearly all noun phrases, not just those that were relevant to either the TE or ST tasks. Competent analysis of coreferent relationships among all noun phrases requires a much more refined knowledge base and much deeper linguistic analysis than we employ in any of our current information extraction components. However, we discovered that 66% of the recall in the CO dry-run test materials was based on references to people and organizations. We also intended to use RESOLVE for coreference resolution within the TE and ST tasks, where it would have to deal with person and organization references.

With these factors in mind, we decided to run RESOLVE on the MUC-6 CO task, but to constrain its input so that it only attempted to find coreferent relationships among references to people and organizations, references that were potentially relevant to the MUC-6 TE and ST tasks. We therefore expected that our recall would be lower than other systems that attempted to find coreferent relationships among the full set of noun phrases defined by the MUC-6 CO task, but we hoped that the evaluation would be valuable nonetheless.

Our official CO scores were 42% recall and 51% precision. If 66% of the recall for the MUC-6 CO final evaluation was based on references to people and organizations, as it was for the dry-run evaluation, then RESOLVE was actually achieving approximately 64% of the total recall for which it was originally intended.[1] After the official evaluation, we retrained RESOLVE based on a prepruning procedure in which only the "easiest" subset of the positive training instances (pairs of coreferent phrases) are used. This version of RESOLVE suffered a small decrease in recall, 39%, but a much larger increase in precision, 59%.

Although our recall and precision results are not among the best reported in this evaluation, we find the results extremely encouraging given the fact that RESOLVE is a fully trainable system. RESOLVE was designed to find coreferent relationships among references to people and organizations; since the MUC-6 CO training material included annotations for entities other than people and organizations, a special interface was used to mark 25 relevant texts from the ST task for person and organization references (5 hours of work). Another reason for specially marking ST texts was a hope that the CO training could also be used for coreference resolution in the TE and ST tasks; unfortunately, time constraints prevented us from using this domain-specific coreference resolution training in the latter two tasks.

We spent 2 weeks on a system component that could reliably identify phrases from BADGER that were relevant candidates for the CO task, which in our view were phrases referring to people and organizations. Another week was spent modifying feature extractors used in the MUC-5 English Joint Ventures domain and adding new features designed specifically for the MUC-6 CO task (especially resolution of pronominal references to people and organizations) and the ST task (especially associating persons with their roles). The output generator, which was required to handle potentially nested COREF SGML tags, required nearly a week-long effort.

Given the nature of trainable decision tree technology, it is safe to assume that RESOLVE's performance would improve for both recall and precision with additional training. We are frankly surprised to see RESOLVE operating as well as it does on the basis of only 25 training documents.

[1] Earlier experiments with RESOLVE in the MUC-5 EJV domain showed that our unpruned C4.5 decision trees used for coreference resolution tend to get higher recall and lower precision than pruned decision trees. Since RESOLVE was already handicapped with respect to potential recall -- focusing only on person and organization references -- we decided to use unpruned trees in the final evaluation.

## Template Entities (TE)

In moving from NE to TE, we add BADGER's processing and a trainable CN dictionary to support BADGER's case frame instantiations. It is interesting to compare our TE scores with our NE scores for organizations and people.

TE is a test of fine-grained information extraction. Not only do we need to locate noun phrases that describe people and organizations, we need to pull those nouns phrases apart in order to separate out names and titles, aliases and locales. This is primarily a noun phrase analysis challenge, with additional points to be won from correct merging and consolidation across multiple noun phrases. As such, the RESOLVE decision trees represent crucial capabilities, along with routines designed to analyze appositives and other complex noun phrases. Unfortunately, CRYSTAL's CN definitions offer little help with noun phrase analysis, since they operate at a relatively coarse level of granularity with respect to a complex noun phrase. The CNs that CRYSTAL induces are designed to locate relevant noun phrases for an extraction task, but they do not help us understand where to look inside a given noun phrase for the relevant information.

**Official TE Score Report**

| SLOT | POS | ACT | COR | PAR | INC | MIS | SPU | NON | REC | PRE | UND | OVG | ERR | SUB |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| organization | 606 | 407 | 378 | 0 | 18 | 210 | 11 | 0 | 62 | 93 | 35 | 3 | 39 | 5 |
| name | 546 | 399 | 239 | 0 | 133 | 174 | 27 | 1 | 44 | 60 | 32 | 7 | 58 | 36 |
| alias | 168 | 121 | 32 | 0 | 10 | 126 | 79 | 236 | 19 | 26 | 75 | 65 | 87 | 24 |
| descriptor | 235 | 14 | 7 | 0 | 1 | 227 | 6 | 236 | 3 | 50 | 97 | 43 | 97 | 13 |
| type | 606 | 407 | 369 | 0 | 27 | 210 | 11 | 0 | 61 | 91 | 35 | 3 | 40 | 7 |
| locale | 114 | 15 | 4 | 0 | 8 | 102 | 3 | 178 | 4 | 27 | 89 | 20 | 97 | 67 |
| country | 115 | 15 | 11 | 0 | 1 | 103 | 3 | 177 | 10 | 73 | 90 | 20 | 91 | 8 |
| person | 496 | 591 | 448 | 0 | 34 | 14 | 109 | 0 | 90 | 76 | 3 | 18 | 26 | 7 |
| name | 496 | 591 | 430 | 0 | 52 | 14 | 109 | 0 | 87 | 73 | 3 | 18 | 29 | 11 |
| alias | 170 | 188 | 140 | 0 | 5 | 25 | 43 | 265 | 82 | 74 | 15 | 23 | 34 | 3 |
| title | 167 | 178 | 158 | 0 | 0 | 9 | 20 | 271 | 95 | 89 | 5 | 11 | 16 | 0 |
| ALL OBJS | 2617 | 1928 | 1390 | 0 | 237 | 990 | 301 | 1364 | 53 | 72 | 38 | 16 | 52 | 15 |
|  |  |  |  |  |  |  |  |  | P&R |  | 2P&R |  | P&2R |  |
| F-MEASURES |  |  |  |  |  |  |  |  | 61.17 |  | 67.29 |  | 56.07 |  |

The irrelevance of CRYSTAL's dictionary to noun phrase analysis has been confirmed by an experimental TE run in which we removed the official CRYSTAL dictionary (containing 2945 CN definitions) and replaced it with a dictionary of only two CN definitions.

> Def1 extracts a person from any syntactic buffer, as long as the people string specialist identified a person in that buffer.
>
> Def2 extracts an organization from any syntactic buffer, as long as the organization string specialist identified an organization in that buffer.

Scores using this dictionary are remarkably close to the official score report based on a fully trained CRYSTAL dictionary, as shown by the following score report. So we must conclude that CRYSTAL, and indeed, all coarse-grained sentence analysis, was totally irrelevant to the TE task. We would have obtained a higher score report by trusting the specialists, working to tighten the performance of the specialists, and focusing on the manually-coded routines needed to dissect noun phrases correctly.

CRYSTAL and BADGER neither helped nor hindered: they just weren't needed. RESOLVE probably did contribute to TE processing, but its positive effect was overwhelmed by critical weaknesses in noun phrase analysis, an area that has not received adequate attention thus far in our quest for trainable technologies.

Here is the TE score report that results from this minimal CN dictionary:

| SLOT | POS | ACT | COR | PAR | INC | MIS | SPU | NON | REC | PRE | UND | OVG | ERR | SUB |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| organization | 606 | 441 | 396 | 0 | 22 | 188 | 23 | 0 | 65 | 90 | 31 | 5 | 37 | 5 |
| name | 546 | 441 | 258 | 0 | 138 | 150 | 45 | 0 | 47 | 59 | 27 | 10 | 56 | 35 |
| alias | 169 | 138 | 39 | 0 | 10 | 120 | 89 | 251 | 23 | 28 | 71 | 64 | 85 | 20 |
| descriptor | 235 | 0 | 0 | 0 | 0 | 235 | 0 | 248 | 0 | 0 | 100 | 0 | 100 | 0 |
| type | 606 | 441 | 386 | 0 | 32 | 188 | 23 | 0 | 64 | 88 | 31 | 5 | 39 | 8 |
| locale | 114 | 0 | 0 | 0 | 0 | 114 | 0 | 184 | 0 | 0 | 100 | 0 | 100 | 0 |
| country | 115 | 0 | 0 | 0 | 0 | 115 | 0 | 183 | 0 | 0 | 100 | 0 | 100 | 0 |
| person | 496 | 593 | 446 | 0 | 33 | 17 | 114 | 0 | 90 | 75 | 3 | 19 | 27 | 7 |
| name | 496 | 593 | 427 | 0 | 52 | 17 | 114 | 0 | 86 | 72 | 3 | 19 | 30 | 11 |
| alias | 170 | 180 | 133 | 0 | 4 | 33 | 43 | 264 | 78 | 74 | 19 | 24 | 38 | 3 |
| title | 167 | 172 | 152 | 0 | 0 | 15 | 20 | 271 | 91 | 88 | 9 | 12 | 19 | 0 |
| ALL OBJS | 2618 | 1965 | 1395 | 0 | 236 | 987 | 334 | 1401 | 53 | 71 | 38 | 17 | 53 | 14 |

| | P&R | 2P&R | P&2R |
|---|---|---|---|
| F-MEASURES | 60.88 | 66.57 | 66.08 |

## Scenario Templates (ST)

### Official TE Score Report

| SLOT | POS | ACT | COR | PAR | INC | MIS | SPU | NON | REC | PRE | UND | OVG | ERR | SUB |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| template | 53 | 64 | 52 | 0 | 0 | 1 | 12 | 35 | 98 | 81 | 2 | 19 | 20 | 0 |
| content | 185 | 175 | 123 | 0 | 6 | 56 | 46 | 0 | 66 | 70 | 30 | 26 | 47 | 5 |
| succession_e | 192 | 197 | 130 | 0 | 6 | 56 | 61 | 0 | 68 | 66 | 29 | 31 | 49 | 4 |
| success_or | 192 | 129 | 65 | 0 | 30 | 97 | 34 | 0 | 34 | 50 | 51 | 26 | 71 | 32 |
| post | 192 | 170 | 63 | 0 | 53 | 76 | 54 | 0 | 33 | 37 | 40 | 32 | 74 | 46 |
| in_and_out | 256 | 193 | 60 | 0 | 22 | 174 | 111 | 0 | 23 | 31 | 68 | 58 | 84 | 27 |
| vac_reason | 192 | 197 | 62 | 0 | 74 | 56 | 61 | 0 | 32 | 31 | 29 | 31 | 75 | 54 |
| in_and_out | 262 | 310 | 164 | 0 | 6 | 92 | 140 | 0 | 63 | 53 | 35 | 45 | 59 | 4 |
| io_person | 262 | 306 | 125 | 0 | 42 | 95 | 139 | 0 | 48 | 41 | 36 | 45 | 69 | 25 |
| new_status | 262 | 304 | 139 | 0 | 31 | 92 | 134 | 0 | 53 | 46 | 35 | 44 | 65 | 18 |
| on_the_job | 262 | 310 | 107 | 0 | 63 | 92 | 140 | 0 | 41 | 35 | 35 | 45 | 73 | 37 |
| other_org | 178 | 5 | 3 | 0 | 2 | 173 | 0 | 58 | 2 | 60 | 97 | 0 | 98 | 40 |
| rel_oth_or | 178 | 5 | 3 | 0 | 2 | 173 | 0 | 58 | 2 | 60 | 97 | 0 | 98 | 40 |
| organization | 113 | 72 | 51 | 0 | 1 | 61 | 20 | 0 | 45 | 71 | 54 | 28 | 62 | 2 |
| name | 110 | 69 | 27 | 0 | 23 | 60 | 19 | 0 | 25 | 39 | 55 | 28 | 79 | 46 |
| alias | 67 | 42 | 12 | 0 | 4 | 51 | 26 | 12 | 18 | 29 | 76 | 62 | 87 | 25 |
| descriptor | 65 | 2 | 0 | 0 | 2 | 63 | 0 | 17 | 0 | 0 | 97 | 0 | 100 | 100 |
| type | 113 | 69 | 50 | 0 | 1 | 62 | 18 | 0 | 44 | 72 | 55 | 26 | 62 | 2 |
| locale | 43 | 7 | 4 | 0 | 3 | 36 | 0 | 17 | 9 | 57 | 84 | 0 | 91 | 43 |
| country | 43 | 7 | 6 | 0 | 1 | 36 | 0 | 17 | 14 | 86 | 84 | 0 | 86 | 14 |
| person | 133 | 138 | 90 | 0 | 6 | 37 | 42 | 0 | 68 | 65 | 28 | 30 | 49 | 6 |
| name | 133 | 138 | 82 | 0 | 14 | 37 | 42 | 0 | 62 | 59 | 28 | 30 | 53 | 15 |
| alias | 85 | 79 | 55 | 0 | 4 | 26 | 20 | 20 | 65 | 70 | 31 | 25 | 48 | 7 |
| title | 81 | 78 | 60 | 0 | 0 | 21 | 18 | 23 | 74 | 77 | 26 | 23 | 39 | 0 |
| ALL OBJS | 2899 | 2285 | 1046 | 0 | 377 | 1476 | 862 | 222 | 36 | 46 | 51 | 38 | 72 | 26 |
| TEXT FILTER | 53 | 64 | 52 | 0 | 0 | 1 | 12 | 35 | 98 | 81 | 2 | 19 | 20 | 0 |

| | P&R | 2P&R | P&2R |
|---|---|---|---|
| F-MEASURES | 40.35 | 43.44 | 37.68 |

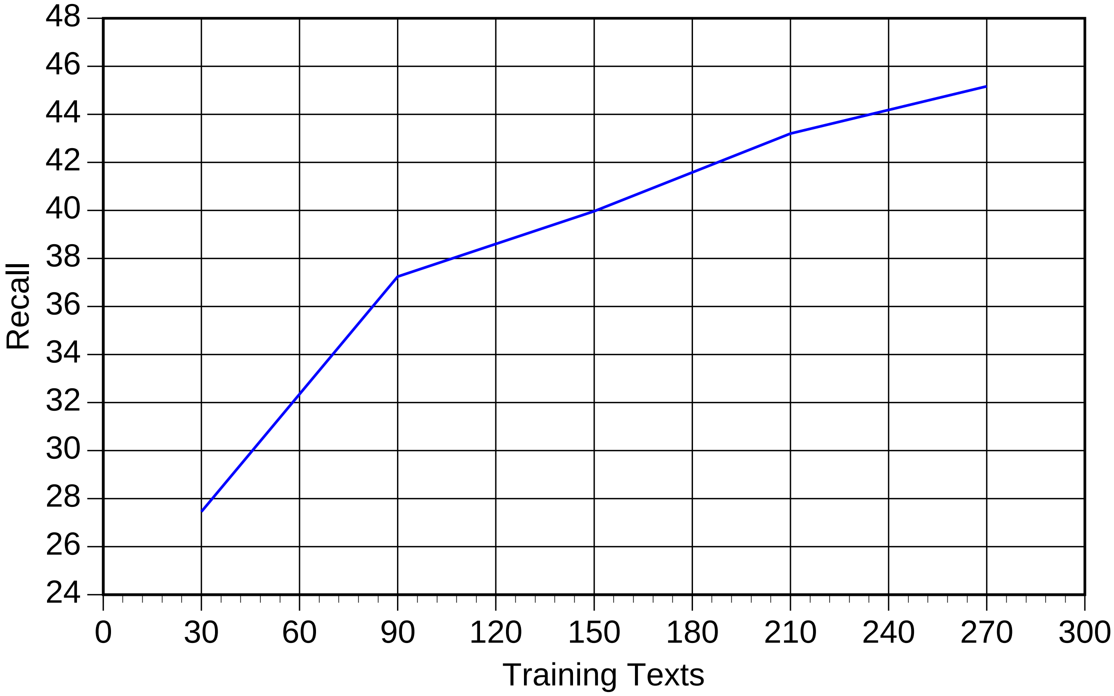
In our system design, the ST task is equivalent to TE plus scenario-specific CNs for changes in position and status and WRAP-UP. WRAP-UP is responsible for all the relational links that describe successions and in-and-out instances. So it is fair to expect that TE will operate as an upper bound for ST. But we see significant drops in both recall and precision as we move from TE to ST.

Part of this drop in recall and precision comes from low recall for new status, which is critical to this scenario. Persons and organizations which are not involved in a change of job status are discarded as irrelevant. CRYSTAL learned CNs named STATUS-IN and STATUS-OUT to identify persons involved in a change of status. The ST training texts provided only about 150 instances each for STATUS-IN and STATUS-OUT, which was insufficient training. CRYSTAL had limited recall for these CNs.

WRAP-UP takes as input the output from BADGER and forms In-and-Out relations and Succession events. WRAP-UP learns to discard as irrelevant any persons and organizations that are not attached to an In-and-Out or a Succession. When our system does not extract a new status due to low recall by domain-specific CNs, this can cause WRAP-UP to discard relevant persons and organizations, further lowering recall. We believe that the training available to CRYSTAL and WRAP-UP was too sparse to enable intelligent inferences about succession events.

The following graph shows the learning curve for CN definitions that identified Organization names. The 300 available TE texts were randomly partitioned into training set and a blind test set with training size ranging from 10% to 90% of the documents. This graph shows the average recall and precision for 50 random partitionings at each training size.

Learning Curve for Organization Name CN definitions:



This data suggests that ORG was probably finding quite a few of the organizations missed by the scanner, but it needed more training. Recall and precision had not leveled out after 270 training texts. With almost no training it got a fourth of the org names (relying heavily on the scanner). After 270 training texts it was getting half the org names, but appears from the slope of the learning curve that it was not reaching saturation yet.

We expect that WRAP-UP would tell a similar story if we were to compute the learning curve for key relational decisions. Inductive learning algorithms eventually flatten out with enough training, but performance tends to increase steadily up until that plateau is reached.

Although CRYSTAL was trained on 300 TE documents, WRAP-UP was only trained on 100 ST documents. The ST training corpus was undoubtedly much too small to support an inductive algorithm designed to learn relational decisions. Trainable technologies are valuable in the battle against the knowledge engineering bottleneck, but we feel that it is important to provide adequate levels of training in order to realize their potential.

## CONCLUSIONS

In the time that has passed since MUC-5 we have been exploring trainable information extraction technologies in a number of different application areas [1, 7]. We have rewritten all of our software in order to enhance cross-platform portability. In addition, we have exploited tagged text documents of the type used in MUC-6. Our experiences with previous MUC evaluations gave us a clear understanding of the hurdles that lie ahead. We made progress in some areas and ignored others completely.

Over the years we have come to appreciate the significant difficulties of software evaluation and the unique problems associated with language processing evaluations. We have pondered the lessons of previous MUCs and worried about the wisdom of a research agenda dominated by score reports. Do performance evaluations succeed in bringing theoretical ideas closer to real-world systems? Or do they inadvertently create a wedge between basic research and applied research?

These big questions are easier to ask than answer, but we have always found the MUC evaluations to be valuable for our own system development efforts. There are always lessons to be learned and a few genuinely new ideas to ponder as well. Two of this year's lessons are painfully obvious to us. Most notably, it pays to engineer the best possible string specialists. If you fail to recognize half of the organizations in a test set, it will be difficult to do well on extraction tasks involving organizations. It is also a mistake to rely on a trainable system component that is not given enough training. We learned that 100 documents do not provide enough training for our system.

Other lessons are more subtle and were not immediately obvious (at least to us). Most notably, the TE task for MUC-6 could be effectively tackled without the benefit of a sentence analyzer. Noun phrase analysis was very important, and coreference resolution played a role, but we saw no benefit from CRYSTAL's dictionary for TE. If the CN definitions had been operating at a finer level of granularity, we might have been able to acquire useful extraction rules for noun phrase analysis. Although our current version of CRYSTAL does not operate at this level, we are currently developing a version of CRYSTAL to learn finer-grained CN definitions.

BADGER's CN output was put to better use in ST where WRAP-UP used CN patterns in order to induce relations between entities. Unfortunately, there was not enough training to produce effective decision trees, so WRAP-UP didn't exactly get a fair trial here. We also noticed that CRYSTAL's dictionary was too sparse to cover all the useful morphological variants for important verbs and verb phrases. In previous MUC evaluations, where 1,000 or more training documents were provided, our dictionary construction tool picked up the more important morphological variants because the training corpus contained examples of them. With only 100 ST training documents this year, our dictionary was missing important verb constructions. For example, the training corpus contained a past tense instance for a specific verb but no present progressive instance. It appears that 100 training texts are not enough for CRYSTAL's dictionary induction algorithm. Or at the very least, a CRYSTAL dictionary based on so little training should be strengthened with recognition routines for morphological variants. These problems do not reflect any essential weakness on the part of CRYSTAL's learning algorithm: they merely illustrate the importance of adequate training materials for machine learning algorithms.

In spite of these difficulties, we are quite pleased with the portability of our trainable system components. The one-month time frame for ST was not a problem for us as far as CRYSTAL, WRAP-UP, and RESOLVE were concerned. CRYSTAL and WRAP-UP could have trained on 1000 texts as easily as 100, and RESOLVE would have been in the same boat if it had been able to train from the MUC-annotated documents directly.

The manual text annotations required for RESOLVE provide us with our final observation about MUC-6 and our MUC-6 system. When a trainable system relies on annotated texts for training, some annotations are more useful than others. Because RESOLVE was designed without any concern for the task definition of the MUC-6 CO task, the training annotations we developed for RESOLVE were not deducible from the annotation system used in MUC-6. As other trainable text processing technologies emerge and develop independent from MUC-6, it may be impossible to create an annotation system that is equally accommodating to all. The inevitable politics of such a situation will be difficult to mediate unless all sites agree to follow the lead of MUC annotation conventions for their own internal development efforts. Although this would ease the problem of diverging systems, it might also suppress imaginative new perspectives on the coreference problem. This conundrum has all the earmarks of a no-win situation.

In summary, our greatest concern after MUC-6 is that the preparation of adequate training corpora may be too expensive or too labor-intensive to enable a fair evaluation of trainable text processing technologies. This will either drive those technologies "underground" (at least with rrespect to MUC meetings), or it may discourage a whole line of research which we feel holds great promise. In our experience, it seems clear that annotated text documents are much less difficult to generate than the key templates used in previous MUC evaluations. It therefore seems ironic to find such a small collection of training documents available to MUC-6 participants this year. We hope that this decision to minimize the training corpus can be reconsidered for future evaluations.

## ACKNOWLEDGEMENTS

This research was supported by NSF Grant No. EEC-9209623, State/Industry/University Cooperative Research on Intelligent Information Retrieval.

## BIBLIOGRAPHY

[1] Aronow, D.B., S. Soderland, J.M. Ponte, F. Feng, W.B. Croft and W.G. Lehnert, “Automated Classification of Encounter Notes in a Computer Based Medical Record”, the 8th World Congress on Medical Informatics, 8:8-12, 1994.

[2] Lehnert, W.G., “Cognition, Computers and Car Bombs: How Yale Prepared Me for the 90’s”, in *Beliefs, Reasoning, and Decision Making: Psycho-logic in Honor of Bob Abelson* (eds: Schank & Langer), Lawrence Erlbaum Associates, Hillsdale, NJ. pp. 143-173. 1994.

[3] Lehnert, W.G., C. Cardie, D. Fisher, J. McCarthy, E. Riloff and S. Soderland, “Evaluating an Information Extraction System,” *Journal of Integrated Computer-Aided Engineering*. 1(6), pp. 453-472. 1995.

[4] Lehnert, W., C. Cardie, D. Fisher, E. Riloff, and R. Williams, “University of Massachusetts: MUC-3 Test Results and Analysis,” in *Proceedings of the Third Message Understanding Conference*, pp. 116-119. 1991.

[5] Lehnert, W., D. Fisher, J. McCarthy, E. Riloff, and S. Soderland, “University of Massachusetts: MUC-4 Test Results and Analysis,” in *Proceedings of the Fourth Message Understanding Conference*, pp. 151-158. 1992.

[6] Lehnert, W., J. McCarthy, S. Soderland, E. Riloff, C. Cardie, J. Peterson, F. Feng, C. Dolan and S. Goldman, “UMass/Hughes: Description of the CIRCUS System as Used for MUC-5,” in *Proceedings of the Fifth Message Understanding Conference*, pp. 277-291. 1993.

[7] Lehnert, W., S. Soderland, D. Aronow, F. Feng, and A. Shmueli, “Inductive Text Classification for Medical Applications”, in *Journal for Experimental and Theoretical Artificial Intelligence (JETAI)*. 7(1), pp. 49-80, 1995.

[8] McCarthy, J., and W. Lehnert, “Using Decision Trees for Coreference Resolution”, in *Proceedings of the International Joint Conference on Artificial Intelligence*, Montreal, Canada. pp. 1050-1055. 1995.

[9] Quinlan, J. Ross. *C4.5: Programs for Machine Learning*, Morgan Kaufmann Pubs., 1993.

[10] Riloff, E. “Automatically Constructing a Dictionary for Information Extraction Tasks”, in *Proceedings of the Eleventh Annual Conference for Artificial Intelligence*. pp. 811-816. 1993.

[11] Riloff, E., and Lehnert, W.G., “A Dictionary Construction Experiment with Domain Experts”, in *Proceedings of the TIPSTER Text Program (Phase I)* , pp. 257-259. 1993.

[12] Riloff, E., and W. Lehnert, “Automated Dictionary Construction for Information Extraction from Text,” in *Proceedings of the Ninth IEEE Conference on Artificial Intelligence for Applications*. IEEE Computer Society Press, pp. 93-99. 1993.

[13] Soderland, W., D. Fisher, J. Aseltine, and W. Lehnert, “CRYSTAL: Inducing a Conceptual Dictionary”, in *Proceedings of the International Joint Conference on Artificial Intelligence*, Montreal, Canada. pp. 1314-1319. 1995.

[14] Soderland, S., and W. Lehnert, “Learning Domain-Specific Discourse Rules for Information Extraction”, in the *Proceedings of the AAAI 1995 Spring Symposium on Empirical Methods in Discourse Interpretation and Generation*, pp. 143-148, Stanford, CA. 1995.

[15] Soderland, S., and W. Lehnert, “Corpus-Driven Knowledge Acquisition for Discourse Analysis”, *Proceedings of the Twelfth National Conference for Artificial Intelligence*. pp. 827-832. 1994.

[16] Soderland, S., and W. Lehnert, “Wrap-Up: a Trainable Discourse Module for Information Extraction”, in *The Journal of Artificial Intelligence Research (JAIR)*. pp. 131-158. 1994.