BBN: Description of the PLUM System as Used for MUC-4

Damaris Ayuso, Sean Boisen, Heidi Fox, Herb Gish, Robert Ingria, and Ralph Weischedel

BBN Systems and Technologies
10 Moulton Street
Cambridge, MA 02138
weisched@bbn.com

BACKGROUND

Traditional approaches to the problem of extracting data from texts have emphasized hand-crafted linguistic knowledge. In contrast, BBN's **PLUM** system (**P**robabilistic **L**anguage **U**nderstanding **M**odel) was developed as part of a DARPA-funded research effort on integrating probabilistic language models with more traditional linguistic techniques. Our research and development goals are

- more rapid development of new applications,
- the ability to train (and re-train) systems based on user markings of correct and incorrect output,
- more accurate selection among interpretations when more than one is found, and
- more robust partial interpretation when no complete interpretation can be found.

A central assumption of our approach is that in processing unrestricted text for data extraction, a non-trivial amount of the text will not be understood. As a result, all components of PLUM are designed to operate on partially understood input, taking advantage of information when available, and not failing when information is unavailable.

We had previously performed experiments on components of the system with texts from the Wall Street Journal, however, the MUC-3 task was the first end-to-end application of PLUM. Very little hand-tuning of knowledge bases was done for MUC-4; since MUC-3, the system architecture as depicted in figure 1 has remained essentially the same.

In addition to participating in MUC-4, since MUC-3 we focused on porting to new domains and a new language, and on performing various experiments designed to control recall/precision tradeoffs. To support these goals, the preprocessing component and the fragment combiner were made declarative; the semantics component was generalized to use probabilities on word senses; we expanded our treatment of reference; we enlarged the set of system parameters at all levels; and we created a new probabilistic classifier for text relevance which filters discourse events.

SYSTEM ARCHITECTURE

The PLUM architecture is presented in Figure 1.

Preprocessing

The input to the system is a file containing one or more messages. The preprocessing module determines message boundaries, identifies the header, and determines paragraph and sentence boundaries. The specification of the input format is now a declarative component of the preprocessor, which enables us to easily digest messages in different formats. This component has proved its utility in porting to two non-MUC formats in the last year.

Morphological Analysis

The first phase of the processing is assignment of part-of-speech information. In BBN's Fast Partial Parser (FPP) [2], a bi-gram probability model, frequency models for known words (derived from large corpora), and heuristics based on word endings for unknown words, assign part of speech to the highly ambiguous words of the corpus. Since these predictions for unknown words were very inaccurate for input that is all upper case, we augmented this part-of-speech tagging with probabilistic models (automatically trained) for recognizing words of Spanish origin and words of English origin. This allowed us to tag new words that were actually Latin American names highly reliably. The Spanish classifier uses a 5 character hidden Markov model, trained on about 30,000 words of Spanish text. The five-gram model of words of English was derived from text from the Wall Street Journal.

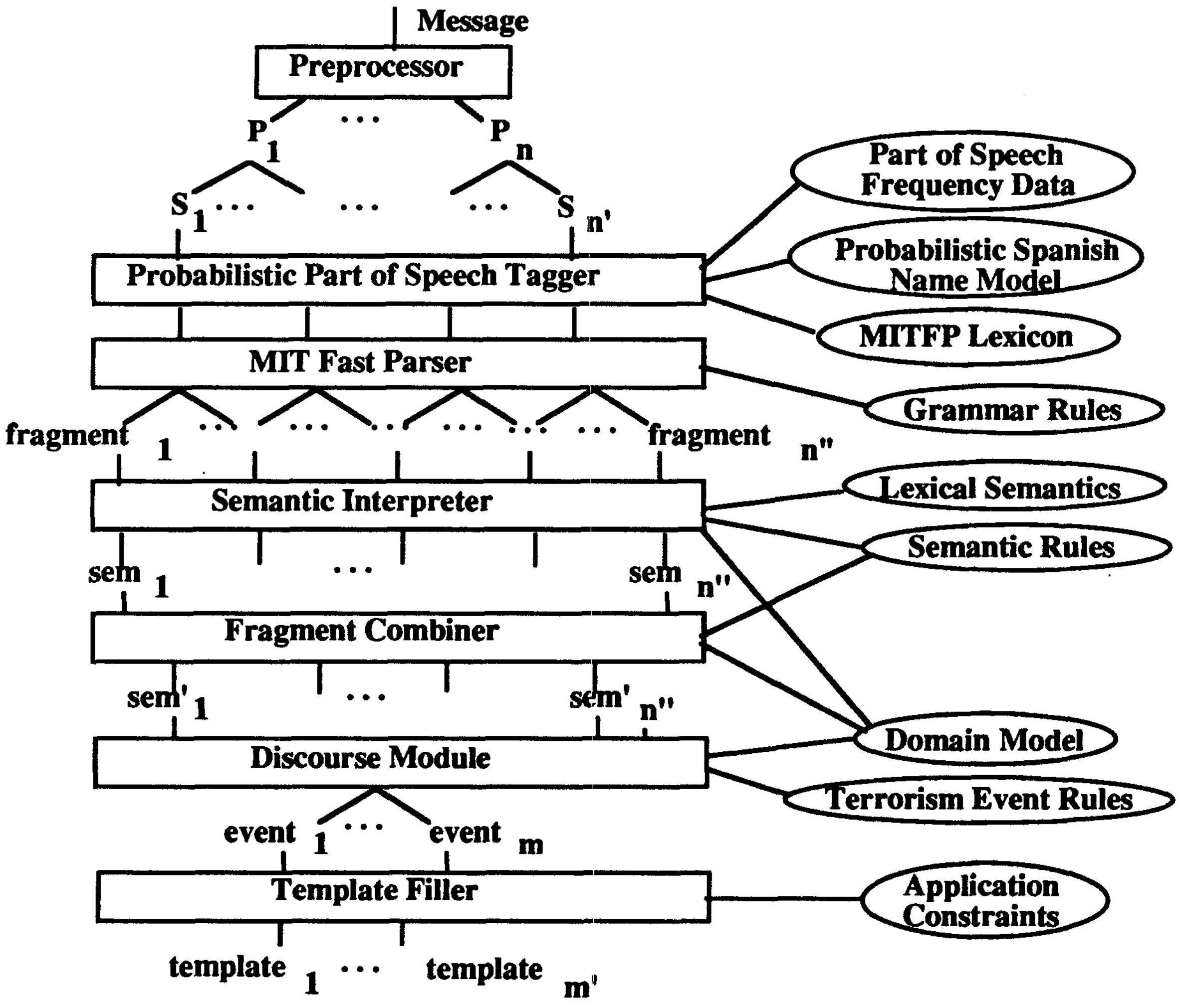

Figure 1. PLUM System Architecture

Parsing

The FPP is a deterministic stochastic parser which does not attempt to generate a single syntactic interpretation of the whole sentence, rather, it generates one or more non-overlapping parse fragments spanning the input sentence, deferring difficult decisions on attachment ambiguities. FPP produces an average of seven fragments for sentences of the complexity seen in the MUC-4 corpus[1].

Here are the 8 parse fragments generated by FPP for the first sentence of TST2-MUC4-0048:

("SALVADORAN PRESIDENT-ELECT ALFREDO CRISTIANI CONDEMNED THE TERRORIST KILLING OF ATTORNEY GENERAL ROBERTO GARCIA ALVARADO"

[1]This number is inflated due to the fact that sentence-final punctuation always appears as a separate fragment, and the fact that commas frequently appear as isolated fragments.

```
(S
 (NP (NP (ADJP (ADJ "SALVADORAN"))
      (N "PRESIDENT-ELECT"))
    (NP (N "ALFREDO" "CRISTIANI")))
 (VP (AUX)
    (VP (V "CONDEMNED")
       (NP (DETERMINER "THE")
          (N "TERRORIST")
          (N "KILLING"))
       (PP (PREP "OF")
          (NP (NP (N "ATTORNEY GENERAL"))
             (NP
              (N (NAME "ROBERTO") (NAME "GARCIA") (NAME "ALVARADO"))))))))
("AND" (CONJ "AND"))
("ACCUSED THE FARABUNDO MARTI NATIONAL LIBERATION FRONT"
 (VP (AUX)
   (VP (V "ACCUSED")
     (NP (DETERMINER "THE")
       (N "FARABUNDO""MARTI""NATIONAL""LIBERATION""FRONT")))))
("(" (PUNCT "("))
("FMLN" (NP (N "FMLN")))
(")" (PUNCT ")"))
("OF THE CRIME" (PP (PREP "OF")
             (NP (DETERMINER "THE")
               (N "CRIME"))))
("." (PUNCT "."))
```

Semantic Interpreter

The semantic interpreter operates on each fragment produced by FPP in a bottom-up, compositional fashion. Throughout the system, defaults are provided so that missing semantic information or rules do not produce errors, but simply mark semantic elements or relationships as unknown. This is consistent with our belief that partial understanding has to be a key element of text processing systems, and missing data has to be regarded as a normal event. The semantic component encompasses both lexical semantics and semantic rules. The semantic lexicon is separate from the parser's lexicon and has much less coverage.

We used an automatic case frame induction procedure to construct an initial version of the lexicon [2]. Word senses of the semantic lexicon have probability assignments, which we plan to derive automatically from corpora. For MUC-4, probabilities were assigned so each word sense is more probable than the next sense of the word as entered in the lexicon.

Lexical semantic entries indicate the word's semantic type (a domain model concept), as well as predicates pertaining to it. For example, here is the lexical semantics for the verb BOMB:

```
(defverb "BOMB" ( BOMB-V-1 BOMBING
                  (:case (subject PEOPLE TI-PERP-OF) (object ANYTYPE OBJECT-OF))))
```

This entry indicates that the type is BOMBING, that a subject argument whose type is PEOPLE should be given the role TI-PERP-OF, and that an object argument of any type should be given the role OBJECT-OF. BOMB-V-1 is the unique identifier of this (only) word sense.

The semantic rules are based on general syntactic patterns, using wildcards and similar mechanisms to provide an extra measure of robustness. The basic elements of our semantic representation are "semantic forms", each of which introduces a variable (e.g. ?13) with a type and a collection of predicates pertaining to that variable.

There are three basic types of semantic forms: entities of the domain, events, and states of affairs. Each of these three can be further categorized as known, unknown, and referential. Entities correspond to the people, places, things, and time intervals of the domain. These are related in important ways, such as through events (who did what to whom) and states of affairs (properties of the entities). Entity descriptions typically arise from noun phrases; events and states of affairs may be described in clauses.

Not everything that is represented in the semantics has actually been understood. For example, the predicate

PP-MODIFIER indicates that two entities (expressed as noun phrases) are connected via a certain preposition. In this way, we have a "placeholder" for the information that a certain structural relation holds between these two items, even though we do not know what the actual semantic relation is. Sometimes understanding the relation more fully is of no consequence, since the information does not contribute to the template-filling task. The information is maintained, however, so that later expectation-driven processing can use it if necessary.

Here is a semantic rule which handles, for example, "group of businessmen", "murder of a man", and "terrorists of the FMLN":

For an NP dominating an NP1, and a PP whose PREP is "OF" and which dominates NP2:

If NP1 is in {"GROUP, "BAND"}	; return semantics of NP2
If NP1 is an EVENT of type TERRORIST	; make NP2 the OBJECT-OF NP1; return new NP1 sem

If type of NP1 is PEOPLE and type of NP2 is ORGANIZATION, merge semantics, showing that NP1 BELONGS-TO NP2; otherwise use a more general NP => NP PP rule

An important consequence of the fragmentation produced by FPP is that top-level constituents are typically more shallow and less varied than full sentence parses. As a result, more semantics coverage was obtained early on in the development process with few semantic rules than would have been expected if the system had had to cover widely varied syntactic structures before producing any semantic structures. In this way, semantic coverage can be added gradually, while the rest of the system is progressing in parallel.

After having assigned semantic representations to the fragments produced by FPP, it is often possible to make some of the attachment decisions which had been deferred. For example, it is possible to combine two NPs of compatible semantic types that are conjoined, or attach prepositional phrases preferentially, using information automatically derived from a corpus [7]. Our basic system uses fragment combination for certain proper name constructions, while some of our submitted optional runs used more extensive patterns for fragment combination. Figure 2 shows a graphical version of the semantics generated for the first fragment of S1 in TST2-MUC4-0048. In this example note the UNKNOWN-EVENT created for the main verb "CONDEMNED", which has no lexical semantics in our system, but still generates a useful semantic representation.

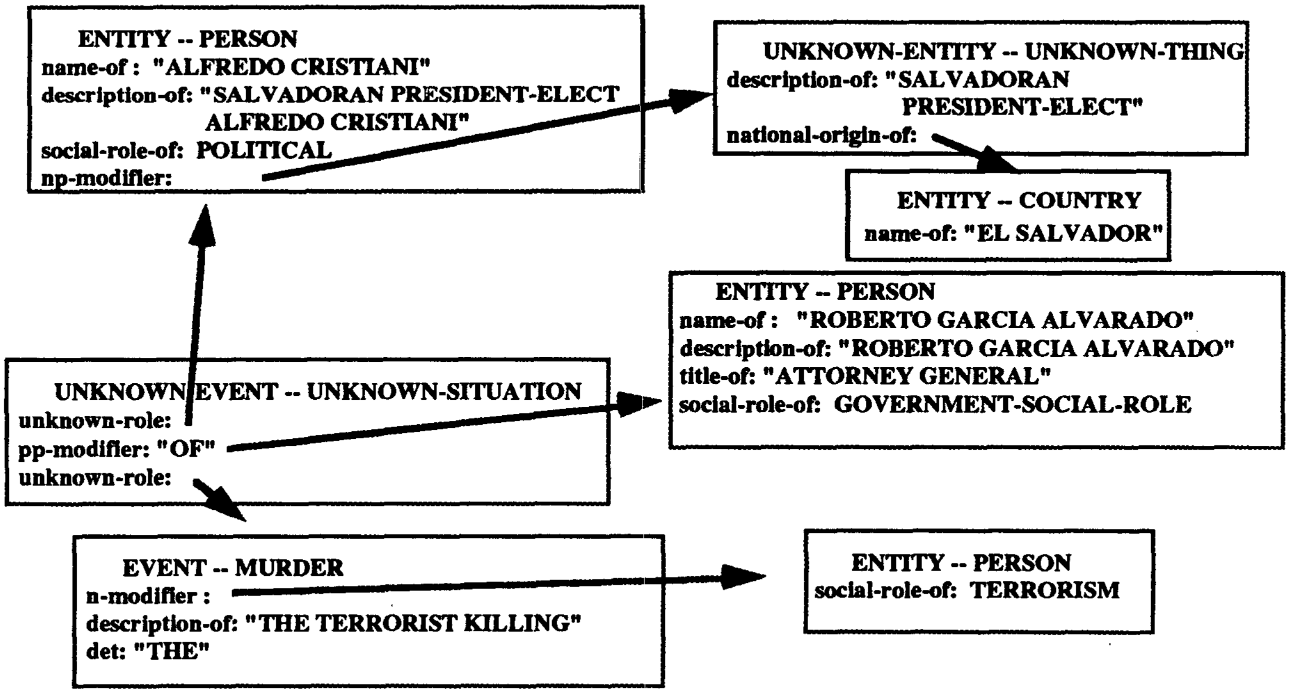
ENTITY -- PERSON
name-of: "ALFREDO CRISTIANI"
description-of: "SALVADORAN PRESIDENT-ELECT ALFREDO CRISTIANI"
social-role-of: POLITICAL
np-modifier:
UNKNOWN-ENTITY -- UNKNOWN-THING
description-of: "SALVADORAN PRESIDENT-ELECT"
national-origin-of:
ENTITY -- COUNTRY
name-of: "EL SALVADOR"
ENTITY -- PERSON
name-of: "ROBERTO GARCIA ALVARADO"
description-of: "ROBERTO GARCIA ALVARADO"
title-of: "ATTORNEY GENERAL"
social-role-of: GOVERNMENT-SOCIAL-ROLE
UNKNOWN-EVENT -- UNKNOWN-SITUATION
unknown-role:
pp-modifier: "OF"
unknown-role:
EVENT -- MURDER
n-modifier :
description-of: "THE TERRORIST KILLING"
det: "THE"
ENTITY -- PERSON
social-role-of: TERRORISM

Figure 2: Example Semantic Representation for "SALVADORAN PRESIDENT-ELECT ALFREDO CRISTIANI CONDEMNED THE TERRORIST KILLING OF ATTORNEY GENERAL ROBERTO GARCIA ALVARADO"

Discourse Processing

The discourse component of PLUM performs the operations necessary to construct event objects corresponding to relevant events in the message. Each event object in the discourse event structure is similar in principle to the notion of a "frame", with its corresponding "slots" or fields. The semantic representation of an event in the text only includes information contained locally in a fragment (after fragment combination); in creating corresponding event objects, the discourse module must infer other long-distance or indirect relations not explicitly found by the interpreter, and resolve any references in the text. The template generator then uses the structures created by the discourse component to generate the final templates. Currently only terrorist incidents (and "possible terrorist incidents") generate discourse events, since these are the core events for MUC-4 template generation. The discourse component was further discussed in [1]. Two primary structures are created by the discourse processor which are used by the template generator: the discourse predicate database and the event structure. The database contains all the predicates mentioned in the semantic representation of the message. When references are resolved, corresponding semantic variables are unified in the database. Any other inferences done by the discourse component also get added to the database.

To create the discourse event structure, the discourse component processes each semantic form produced by the interpreter, adding its information to the database and performing reference resolution when needed. Pronouns and person-type anaphoric definite NPs may be resolved. In addition, set- and member-type reference is also treated in other simpler domains. Some intra-sentential structural constraints on reference are enforced. When a semantic form for an event of interest is encountered, a discourse event is generated, and any slots already found by the interpreter are filled in the event. This event is then merged with a previous event if they are compatible. This heuristic assumes that the events were possibly derived from repeated references to a single real event.

Once all the semantic forms have been processed, heuristic rules are applied to fill in any unfilled slots by looking at text surrounding the forms which triggered a given event. Each filler found is assigned a score based on where it was found in relation to an event trigger, indicating a higher confidence for fillers found closer to a trigger.

Following is the discourse event structure for the first event in TST2-MUC4-0048 :

```
Event: MURDER
Trigger fragments:
  "SALVADORAN PRESIDENT-ELECT ALFREDO CRISTIANI CONDEMNED THE TERRORIST
    KILLING OF ATTORNEY GENERAL ROBERTO GARCIA ALVARADO"
  "GARCIA ALVARADO, 56, WAS KILLED"
  "ITS FRONT GROUPS ARE RESPONSIBLE FOR THE IRRATIONAL
   VIOLENCE THAT KILLED ATTORNEY GENERAL GARCIA"
  ------------------------------------------------------------------
  TI-PERP-OF:              "ITS FRONT GROUPS" (score = 1)
                           "IT" (score = 1) <=> "THE FMLN"
                           "URBAN GUERRILLAS" (score = 2)
  EVENT-TIME-OF:           "JUNE 1" (score = 6)
  EVENT-LOCATION-OF:       "DOWNTOWN SAN SALVADOR" (score = 2)
                           "EL SALVADOR" (score = 2)
  TI-INSTRUMENT-OF:        "A BOMB" (score = 2)
  OBJECT-OF:               "ATTORNEY GENERAL ROBERTO GARCIA ALVARADO" (score = 0)
                            "PRESIDENT OF THE LEGISLATIVE" (score = 2)
                           "AN ARENA LEADER" (score = 2)
```

Each trigger fragment contains one or more words whose semantics triggered this event. In the example above, a score of 0 indicates the filler was found directly by the semantics; 1 that it was found in the same fragment as a trigger semantic form; 2 in the same sentence; 4 in the same paragraph; and 6 in an adjacent paragraph.

Template Generation

The template generator takes the event structure produced by discourse processing and fills out the application-specific templates. Clearly much of this process is governed by the specific requirements of the application, considerations which have little to do with linguistic processing. The template generator must address any arbitrary constraints, as well as deal with the basic details of formatting.

The template generator uses a combination of data-driven and expectation-driven strategies. First the information in the event structure is used to produce initial values. At this point, values which should be filled in but are not available in the event structure are supplied from defaults, either from the header (e.g., date and location information) or from reasonable guesses (e.g. that the object of a murder is usually a suitable filler for the human target slot when the semantic type of the object is unknown).

We expect to eventually use a classifier at this stage of processing. This is especially appropriate for template slots with a set list of possible fillers, e.g. perpetrator confidence, category of incident, etc.

Text Relevance

A new classifier for determining text relevance is now a component of PLUM. It may be utilized by our system to filter out a discourse event object when none of the phrases that gave rise to it is found in a paragraph classified as relevant. Since the event objects are the input to the template generator, it serves effectively as a filter on templates.

The text classifier uses a probabilistic model to perform a binary classification. The features used by the model are stemmed words. The text classifier is trained automatically from two sets of text representing the categories of the classifier (i.e. relevant and irrelevant). A chi-square test is used to determine which words are good indicators of membership on one category but not the other. These words become the features of the probabilistic model. A log probability representing the likelihood of the word occurring in text of one type or the other is assigned to each word. It is this probability that is used in the classification process.

When a piece of text is to be classified, it is scanned for occurrences of the word features selected during training. Summing the log probabilities of all the evidence found in the text gives a measure of the likelihood that the text is a member of a particular category. The sum is then compared to a user-selected threshold to determine the classification. Different thresholds produce different recall and precision values, allowing the user to tune the classifier for high recall, high precision, or something in between. Several of our optional runs showing a wide range of recall-precision tradeoffs were obtained by varying the classifier threshold.

Parameters in PLUM

An important feature of PLUM is that many aspects of its behavior can be controlled by simply varying the values of system parameters. An important goal has been to make our system as "parameterizable" as possible, so that the same software can meet different demands for recall, precision, and overgeneration. PLUM has parameters to control, for example, some aspects of fragment combination, event merging and slot filling by discourse, and relevance assignment by the classifier. In order to pick which system configuration to use for our required MUC-4 run, we tested more than 25 configurations on two test sets and one training set. Maximal F-scores were obtained with settings for aggressively merging events, conservatively looking for slot fillers, and a classifier threshold on relevance of 1.

TEMPLATES FOR EXAMPLE MESSAGE

0. MESSAGE: ID	TST2-MUC4-0048
1. MESSAGE: TEMPLATE	1
2. INCIDENT: DATE	01 JUN 88
3. INCIDENT: LOCATION	EL SALVADOR: SAN SALVADOR (CITY)
4. INCIDENT: TYPE	ATTACK
5. INCIDENT: STAGE OF EXECUTION	ACCOMPLISHED
6. INCIDENT: INSTRUMENT ID	"A BOMB"
7. INCIDENT: INSTRUMENT TYPE	BOMB: "A BOMB"
8. PERP: INCIDENT CATEGORY	TERRORIST ACT
9. PERP: INDIVIDUAL ID	"ITS FRONT GROUPS"
10. PERP: ORG ID	"FMLN"
11. PERP: ORG CONFIDENCE	REPORTED AS FACT: "FMLN"
12. PHYS TGT: ID	-
13. PHYS TGT: TYPE	-
14. PHYS TGT: NUMBER	-
15. PHYS TGT: FOREIGN NATION	-
16. PHYS TGT: EFFECT OF INCIDENT	-

17. PHYS TGT: TOTAL NUMBER	-
18. HUM TGT: NAME	"ROBERTO GARCIA ALVARADO"
19. HUM TGT: DESCRIPTION	"ATTORNEY GENERAL": "ROBERTO GARCIA ALVARADO"
20. HUM TGT: TYPE	GOVERNMENT OFFICIAL: "ROBERTO GARCIA ALVARADO"
21. HUM TGT: NUMBER	1: "ROBERTO GARCIA ALVARADO"
22. HUM TGT: FOREIGN NATION	-
23. HUM TGT: EFFECT OF INCIDENT	DEATH: "ROBERTO GARCIA ALVARADO"
24. HUM TGT: TOTAL NUMBER	-

0. MESSAGE: ID	TST2-MUC4-0048
1. MESSAGE: TEMPLATE	2
2. INCIDENT: DATE	- 19 APR 89
3. INCIDENT: LOCATION	EL SALVADOR: SAN SALVADOR (CITY)
4. INCIDENT: TYPE	BOMBING
5. INCIDENT: STAGE OF EXECUTION	ACCOMPLISHED
6. INCIDENT: INSTRUMENT ID	"THE BOMB" "EXPLOSIVES"
7. INCIDENT: INSTRUMENT TYPE	BOMB: "THE BOMB" EXPLOSIVE: "EXPLOSIVES"
8. PERP: INCIDENT CATEGORY	TERRORIST ACT
9. PERP: INDIVIDUAL ID	"GUERRILLAS"
10. PERP: ORG ID	"FMLN"
11. PERP: ORG CONFIDENCE	REPORTED AS FACT: "FMLN"
12. PHYS TGT: ID	"MERINO'S HOME"
13. PHYS TGT: TYPE	CIVILIAN RESIDENCE: "MERINO'S HOME"
14. PHYS TGT: NUMBER	1: "MERINO'S HOME"
15. PHYS TGT: FOREIGN NATION	-
16. PHYS TGT: EFFECT OF INCIDENT	DESTROYED: "MERINO'S HOME"
17. PHYS TGT: TOTAL NUMBER	-
18. HUM TGT: NAME	-
19. HUM TGT: DESCRIPTION	-
20. HUM TGT: TYPE	-
21. HUM TGT: NUMBER	-
22. HUM TGT: FOREIGN NATION	-
23. HUM TGT: EFFECT OF INCIDENT	DEATH: "-"
24. HUM TGT: TOTAL NUMBER	-

- We correctly identified the 2 events of interest: Merino's murder and the attack on Merino's home.
- We found the correct instruments (though we overgenerated "EXPLOSIVES" in template 2).
- We filled the location, stage of execution, incident category, perpetrator organization correctly in both templates.
- We found the correct perpetrator individual ID in template 2 (and one could argue also in template 1).

However:
- We picked up the wrong dates.
- Our conservative heuristics for picking up multiple targets did not find more than one correct target in these templates.

ACKNOWLEDGEMENTS

The work reported here was supported in part by the Defense Advanced Research Projects Agency and was monitored by Rome Laboratory under Contract No. F30602-91-C-0051. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the United States Government.

REFERENCES

[1] Iwanska, et.al., "Computational Aspects of Discourse in the Context of MUC-3", *Proceedings of the Third Message Understanding Conference (MUC-3)*, May 1991.

[2] Weischedel, R., Ayuso, D. M., Bobrow, R., Boisen, S., Ingria, R., and Palmucci, J. "Partial Parsing, A Report on Work in Progress", *Proceedings of the Fourth DARPA Speech and Natural Language Workshop,* February 1991.