UNIVERSITY OF MARYLAND/CONQUEST: DESCRIPTION OF THE ICTOAN SYSTEM AS USED FOR MUC-4

James Mayfield
Computer Science Department
University of Maryland Baltimore County
Baltimore, MD 21228-5398 USA
mayfield@cs.umbc.edu

INTRODUCTION

The ICTOAN system is a natural language processing system developed jointly by ConQuest, Inc. and the University of Maryland Baltimore County. The system was written from scratch during the first five months of 1992 using an estimated eight person-months of labor. The template generation routines were reused from our MUC-3 system [1], providing leverage of perhaps one person-month. Adaptation of software designed for the ConQuest text retrieval system provided leverage of another six person-months.

The system code was written by the author and by Paul Nelson of ConQuest, Inc. The semantic net representations of world knowledge were developed by Alexander Ho. Roy Cutts, Terri Hobbs, Mark Wilson, and the author wrote the various grammars. Terri Hobbs also cleaned up significant portions of the dictionaries. Paul Riddle modified our MUC-3 template generation software to work with the new template specifications.

We had two main goals in designing the system:

1. to develop a flexible architecture that would support the interleaving of top-down and bottom-up processing.
2. to produce a fast system.

We were largely successful at achieving both of these goals. The ICTOAN system architecture allows low-level and high-level processes to be interleaved and duplicated in arbitrary configurations, which are specified at run-time. And the system is quite fast, processing 100 texts in under twenty minutes.

ARCHITECTURE

The ICTOAN system architecture is based on the idea of multiple parallel streams of data. Each stream carries a particular class of information about the text being processed. For example, a *constituent stream* carries all linguistic constituents found in the text, while an *object stream* carries semantic net nodes representing the meaning of those constituents.

The data in the streams travel in parallel through a pipeline of *processes.* Each process can read items from one or more streams, make any inferences it chooses about those items, and place those same items or new items it creates onto one or more streams.

The processes used for the MUC-4 evaluation first build a semantic net representation of the input story, then fill out templates based on this representation. Three main types of processes were used by ICTOAN to generate a semantic net representation of a MUC-4 input text:

1. **Parsing processes**: these processes attempt to uncover the linguistic structure of the input text.
2. **Disambiguation processes**: these processes reject unlikely interpretations of the input text.
3. **Interpretation processes**: these processes build semantic net structures that represent the meaning of portions of the input text.

The system is designed so that these three types of processes can be intermingled in any desired order. This provides the researcher with an environment in which it is easy to test the effectiveness of a particular process, and affords the system designer great flexibility in tailoring the system to a particular application.

KNOWLEDGE REPRESENTATION

Semantic Nets

ICTOAN uses a semantic net representation language (a variant of the KODIAK knowledge representation language [2]) for meaning representation. Each process has access to the entire semantic net for the story being processed, as well as to the semantic net representing the systems world knowledge. For MUC-4, the world knowledge semantic net contained 3652 concepts.

Dictionaries

ICTOAN used the ConQuest dictionaries for its lexical knowledge. These dictionaries, which were derived from the Proximity Linguistic System, contain 70,000 word senses for 40,000 words with part-of-speech information as well as limited syntactic features.

Grammars

Two types of grammar were used for our evaluation system. Simple context-free grammars with minor augmentations were used for the initial parses. The sentence interpretation component used a grammar that closely resembles a unification grammar (although strict unification is not used). This grammar enforces semantic constraints by verifying that any interpretation to be built meets all constraints expressed in the semantic net. For example, the following rule was used to interpret sentences based on the verb 'assaulted' or the verb 'attacked':

```
(assaulted attacked) {
  syntax s
     subject np
   * verb vp+past+active
     object np
  semantics assault_action
     actor subject
     victim object
}
```

In the section labeled 'syntax,' the names 'subject,' 'verb,' and 'object' are labels given to the semantic representations of the corresponding sub-constituents. The asterisk means that the **vp** is the head of the **s** being built. The section labeled 'semantics' indicates that the semantic net representation of the sentence is an **ASSAULT_ACTION**, for which the **ACTOR** slot is filled by the semantics of the subject, and the **PLACE** slot is filled by the semantics of the object.

STREAMS

The ICTOAN system used three streams during the MUC-4 evaluation:

1. A *constituent stream*, which carried syntactic constituents (*e.g.* noun phrases, prepositional phrases, *etc.*
2. An *object stream*, which carried semantic net nodes representing the meaning of constituents on the constituent stream.
3. An *attack stream*, which carried semantic net nodes that represent attacks described in the story.

The template generator simply observed the attack stream and generated one template for each attack that went by.

PROCESSES

The following five main processes were included in the MUC-4 evaluation system:

- **Statistical word sense disambiguation.** ICTOAN can store multiple word senses for each word in its dictionary. To eliminate some of the ambiguity that arises when processing a word with multiple senses, a statistical process is used to reject some of the less likely senses. This process does a preliminary syntactic parse of each sentence, relative to a fairly complete context-free grammar for sentences. Each word sense is then rated according to the size of the largest constituent that contains it. This information is used during the initial parse to eliminate unlikely parses.

- **Initial parse.** Once word sense has been assigned a likelihood, an initial syntactic parse is done relative to a context-free grammar. This parse is primarily aimed at detecting noun phrases, although in the evaluation system it produced other constituents as well (such as prepositional phrases and verb groups). No semantic information is used at this time; simple features are used to eliminate parses, but they are purely syntactic in nature. The statistics generated by the previous process are used here to eliminate unlikely parses.

- **Phrase interpretation.** The phrase interpreter attempts to build a semantic net representation of each constituent. In the evaluation system, the semantic net was searched for a node with the same name as the head word of the constituent being interpreted. This allowed a wide variety of phrases to be assigned a basic interpretation without a complicated mechanism.

- **Sentence interpretation.** Sentence interpretation is done by using a unification-like grammar to combine the meanings of subconstituents into a single semantic net node representing the meaning of the sentence. This grammar was described in the subsection entitled 'Grammars' above.

- **Template generation.** A template is generated for each attack that passes along the attack stream. The semantic net node representing a particular slot filler is located by traversing a fixed path shape from the node representing the attack. Set fills are then generated by table lookup, while string fills are generated by tracing back from the semantic net node to the longest substring of the input text that has that node as its interpretation.

EXAMPLE

This section describes ICTOAN's processing of the sentence 'GUERRILLAS ATTACKED MERINO'S HOME IN SAN SALVADOR 5 DAYS AGO WITH EXPLOSIVES' from text TST2-MUC4-0048. The initial parsing process first produces a set of possible constituents. Note that some ambiguity remains at this point:

```
[NP: [XNOUNS: GUERRILLAS (UNKNOWN)]]
[VP: [VERB_GROUP: ATTACKED (VERB)]]
[NP: [XNOUNS: MERINO'S (NOUN) HOME (NOUN)]]
[NP: [XNOUNS: MERINO'S (NOUN)]]
[VP: [VERB_GROUP: HOME (VERB)]]
[XPPS: [PP: IN (PREPOSITION)
            [NP: [XPROPERS: SAN SALVADOR (PROPER)]]]]
[NP: [SPECIFIER: [POST_DETERMINER: 5 (NUMBER)]]
     [XNOUNS: DAYS (NOUN)]]
[XADJS: AGO (ADJECTIVE)]
[XADVS: AGO (ADVERB)]
[XPPS: [PP: WITH (PREPOSITION)
            [NP: [XNOUNS: EXPLOSIVES (NOUN)]]]]
. (PUNCT)
```

Next, semantic interpretation is performed on each phrase, and the resulting semantic net nodes are combined by the phrase interpreter. Since the basic rule for the verb 'attack' (shown above in the 'Grammars' subsection) has no provision for the attachment of prepositional phrases, only the subject and direct object are interpreted as part of the resultant **ASSAULT_ACTION**. Here is the structure that is produced:

```
S: [NP: [XNOUNS: GUERRILLAS (UNKNOWN)]
     = GUERRILLAS.198]
   [VP: [VERB_GROUP: ATTACKED (VERB)]]
   [NP: [XNOUNS: MERINO'S (NOUN) HOME (NOUN)]
     = HOME.201]
= ASSAULT_ACTION.203]
```

Finally, a template is generated for this attack:

```
Generating template number 3
  for story TST2-MUC4-0048
  from action node ASSAULT_ACTION.203
Generating string fill for GUERRILLAS.198
String fill selected for GUERRILLAS.198 is "GUERRILLAS"
Generating string fill for GUERRILLAS.198
String fill selected for GUERRILLAS.198 is "GUERRILLAS"
Generating string fill for HOME.201
String fill selected for HOME.201 is "MERINO'S HOME"
```

Here is the resultant template:

```
0.  MESSAGE: ID                    TST2-MUC4-0048
1.  MESSAGE: TEMPLATE              3
2.  INCIDENT: DATE                 -
3.  INCIDENT: LOCATION             -
4.  INCIDENT: TYPE                 ATTACK
5.  INCIDENT: STAGE OF EXECUTION   ACCOMPLISHED
6.  INCIDENT: INSTRUMENT ID        -
7.  INCIDENT: INSTRUMENT TYPE      -
8.  PERP: INCIDENT CATEGORY        TERRORIST ACT
9.  PERP: INDIVIDUAL ID            "GUERRILLAS"
10. PERP: ORGANIZATION ID          "GUERRILLAS"
11. PERP: ORGANIZATION CONFIDENCE  -
12. PHYS TGT: ID                   "MERINO'S HOME"
13. PHYS TGT: TYPE                 CIVILIAN RESIDENCE
14. PHYS TGT: NUMBER               -
15. PHYS TGT: FOREIGN NATION       -
16. PHYS TGT: EFFECT OF INCIDENT   -
17. PHYS TGT: TOTAL NUMBER         -
18. HUM TGT: NAME                  -
19. HUM TGT: DESCRIPTION           -
20. HUM TGT: TYPE                  CIVILIAN
21. HUM TGT: NUMBER                -
22. HUM TGT: FOREIGN NATION        -
23. HUM TGT: EFFECT OF INCIDENT    -
24. HUM TGT: TOTAL NUMBER          -
```

References

[1] James Mayfield and Edwin Addison. Synchronetics: Description of the Synchronetics system used for MUC-3. In Beth Sundheim, editor, *Proceedings of the Third Message Understanding Conference (MUC-3)*, pages 207–211. Morgan Kaufmann, 1991.

[2] Robert Wilensky. Some problems and proposals for knowledge representation. Memorandum UCB/CSD 87/351, University of California, Berkeley Electronic Research Laboratory, 1987.