HUGHES RESEARCH LABORATORIES: DESCRIPTION OF THE TRAINABLE TEXT SKIMMER USED FOR MUC-4

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

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

INTRODUCTION

The objective of the Hughes Trainable Text Skimmer (TTS) Project is to create text skimming software that: (1) can be easily re-configured for new applications, (2) improves its performance with use, and (3) is fast enough to process several megabytes of text per day. The TTS-MUC4 system is our second full-scale prototype. It is an adaptation of the TTS-MUC3 system [1] [2], which constituted our first-full scale text skimming prototype.

TTS-MUC3 utilized a previously-constructed text database facility and pattern matcher used for shallow parsing. Its modular process model integrated the results of case memory retrieval over sentences from multiple stories, extracting the date and location of incidents, and computing cross-reference information for various slots. One calendar month and approximately three (3) person months were spent developing TTS for MUC-3.

TTS-MUC3 demonstrated that a pattern classification approach was promising for performing text skimming. TTS-MUC4 is similar to TTS-MUC3, with a few minor changes. First, the K-Nearest Neighbor classifier used in TTS-MUC3 was replaced in TTS-MUC4 with a Bayesian Classifier which actually includes specialized classifiers for each slot. Therefore, for :INCIDENT-TYPE the set of features present in an entire sentence were used as features, but for :HUM-TGT-NAME the features just before and after a candidate were used. Secondly, in the new prototype, code was added to extract information to fill the new and revised slots of the MUC-4 templates. Thirdly, additional filters were developed to improve the precision of the values of the template fillers. Like our first prototype, TTS-MUC4 incorporates semi-automated lexicon generation and almost fully automated phrase pattern generation. Two calendar months and approximately 2.5 person months were spent on enhancing the TTS-MUC3 system to create TTS-MUC4.

As with TTS-MUC3, all the modules in TTS-MUC4 are domain independent. All the modules except the date and location extraction modules are trained prior to skimming. In addition, the location extraction module requires a location database, including the specification of which locations are contained within others.

The goal of the TTS project is to develop a text skimmer that can be used in a variety of applications. By relying on statistical information processing and keeping the amount of domain-dependent information to a minimum, it is hoped that this system can be easily ported to a variety of tasks, such as analysis of finance-related wire-service stories.

THE TTS APPROACH

There are two aspects of TTS. The first is system training, or the process of deriving or identifying the phrases which are used in the training corpus. The second is text skimming, or the process of skimming a text with the purpose of identifying whether the text falls into a specific category, and if so, extracting particular pieces of information from the text. Figure 1 illustrates the key components of both aspects of Hughes Trainable Text Skimmer. The training involves deriving the set of phrases used to generate the templates associated with a particular corpus of texts, and the generation of training features which map features to the actual text in the stories. The text skimming involves the Text Database, databases containing derived phrases and training features, a Phrasal Parser, the Classifier, and a Feature-to-Template Process Model.

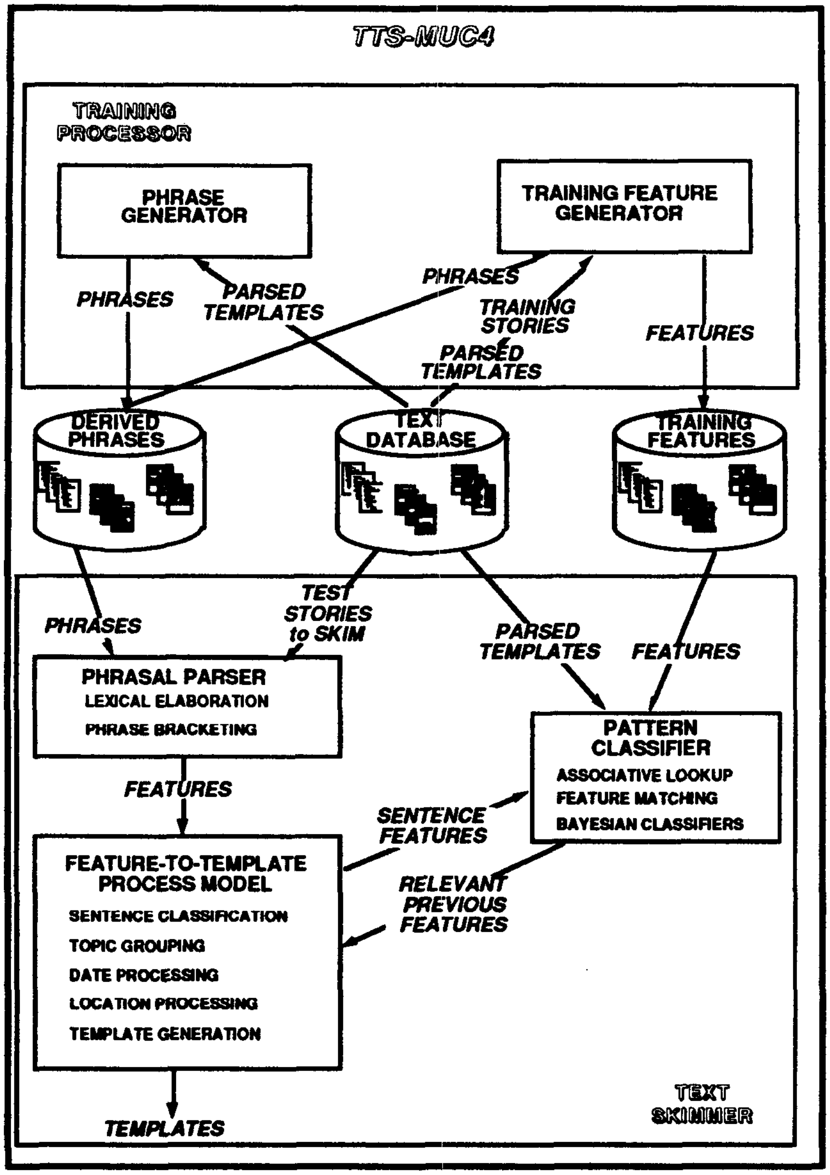

Figure 1: A block diagram of Hughes Trainable Text Skimmer.

TTS Module Descriptions

The ***training processor*** uses the provided templates to derive the phrase lexicon and build the features used by the pattern classifier. Phrases are generated from the fillers for the template slots. Features are generated from the sentences that provided the fillers. The word lexicon is generated by performing a word frequency analysis on the raw text. In TTS-MUC4, all words that occur between 10 and 105 times are included in the lexicon.

The ***text database*** contains: (1) the database of training stories, (2) the database of testing stories, (3) the database of training templates for user browsing, and (4) the database of parsed templates for use during training. It supports retrieval of a single fragment of text from a large collection of texts that may be spread over multiple disk files. The values retrieved from the text database fall into three categories. A raw text string can be retrieved for processing a story or browsing templates. A recursive token structure, representing an entire story, with individual words at the leaf nodes, can also be returned. In the third case, an s-expression representing a parsed template can be retrieved for further use by the pattern classifier.

The ***phrasal parser*** is a fast, shallow, conceptual parser. The parser accepts a token structure, a lexical hierarchy, and a phrase-pattern set. The parser returns an ordered list of text features. A text feature includes: (1) a member of the concept hierarchy, (2) the string covered by the phrase, and (3) a recursive token structure spanning the tokens covered by the phrase.

Lexicon entries are created by adding word stems to a concept hierarchy as follows,

```
(ks:isa h-lex "PRIEST"      :religious-individual-w)
(ks:isa h-lex "MISSIONARY"  :religious-individual-w)
(ks:isa h-lex "CONFERENCE"  :conference-w)
(ks:isa h-lex "SUMMIT"      :conference-w)
(ks:isa h-lex "RECEPTION"   :conference-w)
```

Phrasal pattern definitions have three parts: the discrimination net to which the pattern belongs, a list of the pattern components, and the pattern descriptor. Phrasal patterns may reference either elements of the concept hierarchy, or specific words. Pattern definitions have the syntax illustrated in the following examples:

```
(ph:defpattern (net ? h-con)
      (:determiner :small-number :unidentified-w :human-group-w)
      :civilian  )
(ph:defpattern (net ? h-con)
      (:civilian-w "FROM" :number-w :spanish-name-w "AREA")
      :civilian  )
(ph:defpattern (net ? h-con)
      (:public-w :communication-device-w :building-w)
      :communications  )
```

The features are extracted using a depth first search of the patterns, with a preference for patterns that have specific words over those which have only concept names, as well as a preference for longer patterns.

The ***pattern classifier*** actually consists of a separate pattern classifier for each template slot to be filled. Each such classifier takes an ordered list of text features, and returns the probability of a set fill or a string fill, based upon all previously used fills for that particular slot.

The ***feature-to-template process model*** has the task of identifying which features of the test text are relevant to each template slot. It also has the task of generating a completed story template for each relevant topic identified in the story.

Flow of Control

Once an initial training phase has been completed to initialize the pattern classifiers, the feature-to-template process model performs its task in four phases: (1) pattern classification, (2) topic grouping, (3) slot filling, and (4) template generation. During the first phase, TTS-MUC4 iterates over all sentences in the text and collects potential topics. The second phase consists of determining which topics are relevant, and eliminating from further consideration the sentences having no bearing on the chosen topics. In phase 3, the values to be supplied in the completed template(s) are extracted and/or computed from the relevant sentences, based upon the focus of each selected topic. Lastly, the standard MUC-4 templates are generated.

Pattern Classification

For each sentence of a story, a set of Bayesian classifiers is used. For set fills, the classifiers compute the probabilities of the different potential set fills. For string fills, the classifiers compute the probability that a particular phrase is, for example, a human target. (See the Site Report section of this volume for a detailed description of TTS-MUC4, including details on the Bayesian classifiers.)

Topic Grouping and Relevance Assessment

Topic grouping (analogous to discourse processing) is based on the INCIDENT: TYPE slot. The weight for each type of incident is computed for every sentence. The weights are then passed through a competitive filter, resulting in binary signals. The competitive filter first normalizes the topic weights using a Gaussian mask on a sentence by sentence basis, then computes the best topic. A topic is a set of contiguous sentences with the same computed value for INCIDENT: TYPE.

Figure 2 shows the inputs and outputs to the topic grouping process. Note that moderately high evidence of kidnapping throughout the story is suppressed in favor of the bombing interpretation, which turns out to be correct. This filter used is topic grouping is designed to pick out signals that are high but that "drop out" from time to time, as one can see in the smoothing over the arson signal.

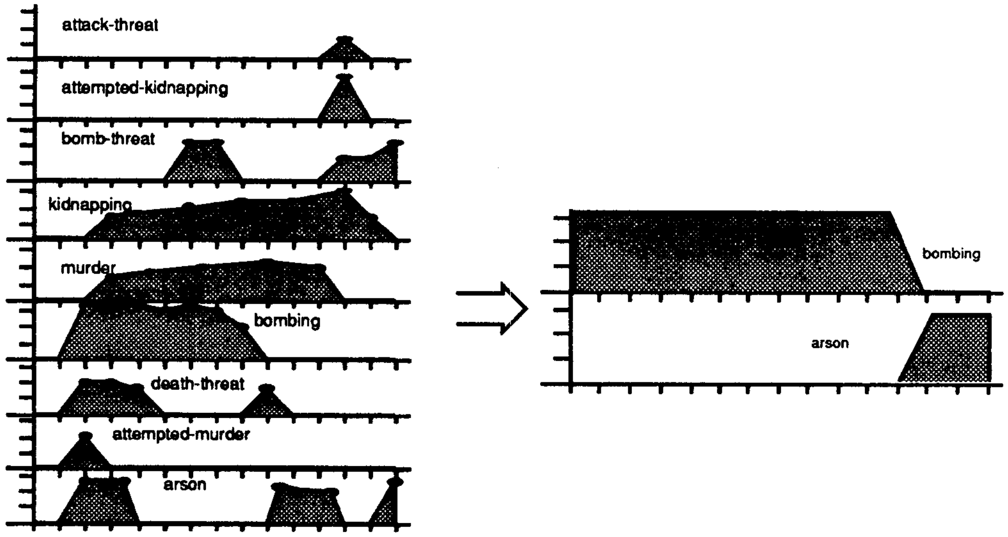

Figure 2: Input and output to topic grouping for TST-MUC3-0099.

Slot Filling

Slot filling consists of five parts: (1) pure set fills, (2) string fills, (3) cross-referenced slots, (4) date extraction, and (5) location extraction. The first three parts consider only relevant sentences. A relevant sentence shares the same topic with the previous sentence or contains no competing topic. There are two distinct types of processing for slot filling. Two slots, date and location, are filled by domain specific procedures. The remaining slots are filled using hypotheses returned by the pattern classifier. The slots are filled from the pattern classifier fall into three categories:

1. Set fills—Pure set fills are computed by averaging the weights over all sentences for a given topic, and picking the highest score.

2. String fills—String fills are computed in a similar manner to set fills. They differ in that the suggestions returned by the classifier are subject to a threshold on the weights. For the official run of TTS-MUC4 the string fill threshold was set at 0.75.

3. Cross reference generation—Cross reference generation is performed by choosing the most likely tag (as suggested by the classifier) for the sentence that contains the string fill.

For date extraction, all sentences within a topic are scanned for absolute or relative date references. Absolute date references are combined into a range. Absolute dates are preferred over relative dates within a given sentence. Relative date references are interpreted with respect to either the current date specification for a story (if one has been found) or the story date line.

For location extraction, all sentences within a topic are scanned for known location names. The resulting list of location names is then searched for a maximal, legal, location containment chain.

TST-MUC4-0048 ANALYSIS

A detailed look at the processing of TST-MUC4-0048 provides insight into TTS-MUC4. Upon reading the first sentence of TST-MUC4-0048:

```
SALVADORAN PRESIDENT-ELECT ALFREDO CRISTIANI CONDEMNED THE
TERRORIST KILLING  OF ATTORNEY GENERAL ROBERTO GARCIA
ALVARADO AND ACCUSED THE FARABUNDO MARTI NATIONAL LIBERATION
FRONT (FMLN) OF THE CRIME.
```

TTS-MUC4 extracts the following features:

```
(feature
   ((feature :of-country-w "SALVADORAN"
             #<token:SALVADORAN>))
   ((feature :politician-w "PRESIDENT"
             #<token:PRESIDENT>))
   ((feature :elect-w "ELECT"
             #<token:ELECT>))
   ((feature :spanish-name "ALFREDO CRISTIANI"
             #<token-spanish-name>))
   ((feature :condemn-w "CONDEMNED"
             #<token:CONDEMNED>))
   ((feature :terrorist-act-indiv "THE TERRORIST"
             #<token-terrorist-act-indiv>))
   ((feature :death-w "KILLING"
             #<token:KILLING>))
   ((feature :government-official-or-legal-or-judicial-descr
             "ATTORNEY GENERAL"
             #<token-government-official-or-legal-or-judicial-descr>))
   ((feature :spanish-name "ROBERTO GARCIA ALVARADO"
             #<token-spanish-name>))
   ((feature :doubt-1-w "ACCUSED"
             #<token:ACCUSED>))
   ((feature :terrorist-act-org
             "THE FARABUNDO MARTI NATIONAL LIBERATION FRONT (FMLN)"
             #<token-terrorist-act-org>))
   ((feature :crime-w "CRIME" #<token:CRIME>)))
```

TTS-MUC4 identifies each potentially relevant template slot, and hypothesizes possible values, based upon similar sentences which TTS-MUC4 has retrieved from case memory. In this case, for example, the system chooses `:attack` as the most likely `:incident-type`, based upon the following calculated likelihoods for each topic associated with the first sentence of -0048:

```
(:incident-type
     (:attack . 0.9828392566462706)    (:robbery . 0.0)
     (:kidnapping . 0.0)    (:bombing . 5.191340765556957E-4)
     (:arson . 0.0))
```

Similar likelihoods are calculated for each sentence.

Based on semantic features such as `:death-w`, `:government-official-or-legal-or-judicial-descr`, and `:terrorist-act-org`, the Bayesian classifier for `INCIDENT-TYPE` computes the probabilities above. Likewise, using features such as `:death-w` and `:terrorist-act-org`, another Bayesian classifier computes the probability that "`ROBERTO GARCIA ALVARADO`" is a human target.

Processing proceeds in a like manner for the next 6 sentences of the story, and the template shown in figure 3 is produced.

```
0.  MESSAGE: ID                         TST2-MUC4-0048
1.  MESSAGE: TEMPLATE                   1
2.  INCIDENT: DATE                      01 JUNE 1988
3.  INCIDENT: LOCATION                  EL SALVADOR: SAN SALVADOR (DEPARTMENT)
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                 "URBAN GUERRILLAS"
10. PERP: ORGANIZATION ID               "NATIONALIST REPUBLICAN ALLIANCE"
11. PERP: ORGANIZATION CONFIDENCE       SUSPECTED OR ACCUSED:
                                           "NATIONALIST REPUBLICAN ALLIANCE"
12. PHYS TGT: ID                        "VEHICLE"
13. PHYS TGT: TYPE                      OTHER: "VEHICLE"
14. PHYS TGT: NUMBER                    1: "VEHICLE"
15. PHYS TGT: FOREIGN NATION            *
16. PHYS TGT: EFFECT OF INCIDENT        DESTROYED: "VEHICLE"
17. PHYS TGT: TOTAL NUMBER              1
18. HUM TGT: NAME                       "GARCIA ALVARADO"
19. HUM TGT: DESCRIPTION                "DEMOCRAT": "JOSE NAPOLEON DUARTE"
                                        "ATTORNEY GENERAL":
                                           "ROBERTO GARCIA ALVARADO"
20. HUM TGT: TYPE                       CIVILIAN: "ROBERTO GARCIA ALVARADO"
                                        CIVILIAN: "JOSE NAPOLEON DUARTE"
                                        CIVILIAN: "GARCIA ALVARADO"
21. HUM TGT: NUMBER                     1: "GARCIA ALVARADO"
                                        1: "DEMOCRAT"
                                        1: "ATTORNEY GENERAL"
22. HUM TGT: FOREIGN NATION             *
23. HUM TGT: EFFECT OF INCIDENT         DEATH: "GARCIA ALVARADO"
                                        DEATH: "ATTORNEY GENERAL"
                                        DEATH: "DEMOCRAT"
24. HUM TGT: TOTAL NUMBER               5
```

Figure 3: Template produced for TST-MUC4-0048 sentences 1-7.

The most notable feature of this template fill is over generation. TTS-MUC4 correctly identifies one human target and one physical target, but one other person, JOSE NAPOLEON DUARTE, and several coreferents are also generated.

Sentences 11 through 13 of TST-MUC4-0048:

> GUERRILLAS ATTACKED MERINO'S HOME IN SAN SALVADOR 5 DAYS AGO
> WITH EXPLOSIVES. THERE WERE SEVEN CHILDREN, INCLUDING FOUR
> OF THE VICE PRESIDENT'S CHILDREN, IN THE HOME AT THE TIME.
> A 15-YEAR-OLD NIECE OF MERINO'S WAS INJURED.

along with the preceding 2 sentences and the subsequent sentence 14, produce the template shown in figure 4.

```
0.  MESSAGE: ID                           TST2-MUC4-0048
1.  MESSAGE: TEMPLATE                     2
2.  INCIDENT: DATE                        - 19 APR 1989
3.  INCIDENT: LOCATION                    EL SALVADOR: SAN SALVADOR (DEPARTMENT)
4.  INCIDENT: TYPE                        BOMBING
5.  INCIDENT: STAGE OF EXECUTION          ACCOMPLISHED
6.  INCIDENT: INSTRUMENT ID               "EXPLOSIVES"
7.  INCIDENT: INSTRUMENT TYPE             OTHER: "EXPLOSIVES"
8.  PERP: INCIDENT CATEGORY               TERRORIST ACT
9.  PERP: INDIVIDUAL ID                   "GUERRILLAS"
10. PERP: ORGANIZATION ID                 *
11. PERP: ORGANIZATION CONFIDENCE         *
12. PHYS TGT: ID                          "MERINO'S HOME"
                                          "THE VEHICLE"
13. PHYS TGT: TYPE                        TRANSPORT VEHICLE: "THE VEHICLE"
                                          POLITICAL FIGURE OFFICE OR RESIDENCE:
                                             "MERINO'S HOME"
14. PHYS TGT: NUMBER                      1: "MERINO'S HOME"
                                          1: "THE VEHICLE"
15. PHYS TGT: FOREIGN NATION              *
16. PHYS TGT: EFFECT OF INCIDENT          SOME DAMAGE: "THE VEHICLE"
                                          (DESTROYED SOME DAMAGE): "MERINO'S HOME"
17. PHYS TGT: TOTAL NUMBER                2
18. HUM TGT: NAME                         "MERINO'S"
19. HUM TGT: DESCRIPTION                  "VICE PRESIDENT'S"
                                          "CHILDREN"
                                          "DRIVER"
                                          "CHILDREN"
20. HUM TGT: TYPE                         CIVILIAN: "CHILDREN"
                                          CIVILIAN: "DRIVER"
                                          CIVILIAN: "CHILDREN"
                                          CIVILIAN: "VICE PRESIDENT'S"
                                          CIVILIAN: "MERINO'S"
21. HUM TGT: NUMBER                       1: "MERINO'S"
                                          1: "VICE PRESIDENT'S"
                                          1: "CHILDREN"
                                          1: "DRIVER"
                                          7: "SEVEN CHILDREN"
22. HUM TGT: FOREIGN NATION               *
23. HUM TGT: EFFECT OF INCIDENT           INJURY: "MERINO'S"
                                          DEATH: "SEVEN CHILDREN"
                                          DEATH: "DRIVER"
                                          DEATH: "CHILDREN"
                                          DEATH: "VICE PRESIDENT'S"
24. HUM TGT: TOTAL NUMBER                 11
```

Figure 4: Template produced for TST-MUC4-0048 sentences 9-14.

The most notable feature of these template fills is the fragmentation of the string fills. Many of the correct features are tagged, but, because TTS only has a shallow parser, the phrases: "CHILDREN" and "VICE PRESIDENT'S" are never combined into the correct answer "VICE PRESIDENT'S CHILDREN."

In addition to the two templates above, TTS-MUC4 also produces a spurious template based on the sentences 21 through 22:

```
ACCORDING TO THE POLICE AND GARCIA ALVARADO'S DRIVER, WHO
ESCAPED UNSCATHED, THE ATTORNEY GENERAL WAS TRAVELING WITH
TWO BODYGUARDS.  ONE OF THEM WAS INJURED.
```

and the subsequent 3 sentences. The template produced is not spurious, in a sense, because it describes the same incident as template 1. However, because TTS-MUC4 does not have a full discourse processing component, the two templates are not combined.

CONCLUSIONS

To understand the performance of TTS-MUC4, one should look at the the inter-dependence between the various processing modules. Figure 5 shows these dependencies. Each module points to the modules upon which it depends. We contend that improving a module will enable improvement of the behavior of its dependents.

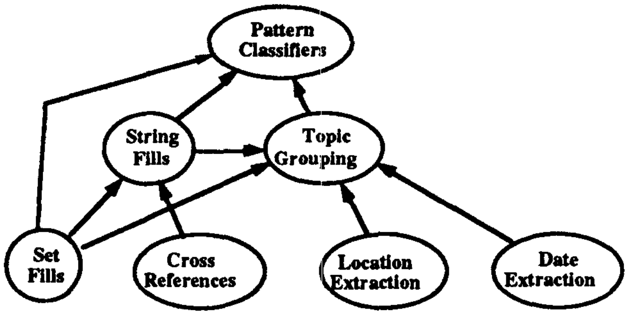

Figure 5: Module dependency graph.

For example, in tuning the system parameter, the pattern classifier for each slot was adjusted to balance recall and performance. In every case, adjusting that parameter to optimize the pattern classification resulted in increased performance of the overall system.

REFERENCES

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

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