MUCking In, or Fifty Years in Information Extraction

Ralph Grishman

New York University, Professor Emeritus of Computer Science
grishman@cims.nyu.edu

I want to thank the ACL for this Lifetime Achievement Award. I am deeply honored to be receiving it. I would also like to thank the students, faculty, and researchers who were members of the Proteus Project during most of my professional lifetime. It was an honor to serve that group.

1. Introduction

Those of us who are fortunate enough to look back on a long, fulfilling lifetime are sometimes faced with limitations at this stage. Mine is Parkinson's disease, which is a neurological condition marked by involuntary movements that you may notice as I speak. Due to a secondary illness on top of the Parkinson's disease, I am currently unable to give a full presentation to the ACL members today. Fortunately, the Proteus team has stepped forward in my stead and I am happy to pass the talk on to one of its chief members, Adam Meyers. Dr. Meyers is a professor and researcher at New York University and has worked on information extraction, particularly on its tool development, for the past 20 years. I am happy to pass the delivery of my talk over to him . . . Adam?

2. Starting Out

My first contact with computing was in a summer 1964 course in Fortran and assembly language programming at New York University (NYU) when I had just graduated from high school. In September I was offered a research slot by Jack Schwartz for four hours a week to write a simulator for a parallel computer, the Control Data 6600. This job continued while I attended Columbia as a physics major.

While still working at NYU, its Courant Institute enabled me to publish—all on state-of-the-art mimeograph paper—a text on assembly language programming for the Control Data computer. The text was formally published by Algorithmics Press in 1971 (Grishman 1971) as *Assembly Language Programming for the Control Data 6000 Series.* Because this was the only text on assemblers, it went into two more editions.

I graduated from Columbia College in Spring 1968. I had wanted to go to Switzerland after graduation, so in the spring of 1968 I wrote to the handful of companies there that had a Control Data computer, asking if they needed help breaking in a CDC 6600 computer. This allowed me to spend a productive summer in Zurich at the Fides Union Fiduciaire.

https://doi.org/10.1162/coli_a_00547

I continued my studies in physics as a grad student while teaching an introductory programming course—I think in PL1—at Barnard and working on parallel computing with Jack Schwartz as well.

3. The Linguistic String Project and its Goal

While engaged in all this work, Jack introduced me to Naomi Sager, thinking I might be a good match for the Fortran programmer she needed for her Linguistic String Project.

Naomi had come to NYU in 1965 with a master's degree in Electrical Engineering from Columbia, five years working as an electronics engineer in the Biophysics Department at Memorial Sloan Kettering Cancer Center, and five more years at Penn where she worked with Zellig Harris on "natural language computer processing." There she was part of the team that developed the first English language parsing program, running on the UNIVAC I. I believe she then came to NYU because it was the first non-governmental institution to have a UNIVAC.

Naomi's goal in natural language processing was to analyze patient discharge summaries and extract information from them that could be used in medical applications, and my future in natural language processing (NLP) began with this goal.

4. Information Extraction[1]

Information extraction (IE) is the process of identifying within text instances of specified classes of entities and of predications involving these entities. An early and oft-cited example is the extraction of information about *management succession*—executives starting and leaving jobs.[2] If we were given the text

> Fred Flintstone was named CTO of Time Bank Inc. in 2031. The next year he got married and became CEO of Dinosaur Savings & Loan.

we would produce the table seen in Figure 1.

A few things to note about this table:

- Information about his marriage was not captured above; extraction seeks in general to cover only a predefined set of predications.
- The second sentence refers to "he," but that would not be very meaningful in the table; we resolve "he" to the name of the entity being referenced, "Fred Flintstone."
- The two events use quite different choices of words to express the event ("was named" and "became"), but the extraction procedure recognizes these as instances of the same type of event. Such recognition of paraphrases is a critical aspect of information extraction.

Why do information extraction? A short answer is that it makes the information in the text *more accessible* for further processing. Getting a person's employment history from

[1] This section is taken from Grishman (2012).

[2] This is a simplified version of the task for Message Understanding Conference-6, to be discussed below.

person	company	position	year	in/out
Fred Flintstone	Time Bank Inc.	CTO	2031	in
Fred Flintstone	Time Bank Inc.	CTO	2032	out
Fred Flintstone	Dinosaur Savings & Loan	CEO	2032	in

Figure 1
A simple example of IE.

a large collection of news articles is a complex procedure; getting it from a table such as that shown above is far easier. Similarly, searching for an interaction of two particular genes in the scientific literature will be much easier if these relations have been extracted in advance. This is not a new idea. Zellig Harris, over 50 years ago, described how one might identify the basic semantic structures in a field of science, and then map scientific articles into these structures (Harris 1958). These structures could then be used for literature searches. While we have made progress in IE (and in NLP more generally) in these 50 years, we are still some ways away from fully realizing Harris's vision.

5. Tool Development for the String Project

In the year after I joined the String Project, I completed a Fortran parser and we started doing experiments with it. The first experiments used simple rules for building structures from the words and gradually we built more complex rules for the more complex structures.

The simplest rules involved linear sequences of words. The more complex constructions, for example, passive verb constructions, reduced relative verb forms like "The patient examined by the doctor," and nominalizations like "the doctor's examination of the patient"—this, of course, gave us better coverage.

Over the years, a large set of tools has been built up to support our IE work. Many of these tools are freely available on the Proteus Project site.

Both the Linguistic String Project and the Proteus Project have held to the idea that IE should begin with a full syntactic analysis followed by a semantic analysis of the syntactic structure.[3] Our commitment to the full syntactic analysis has led to the creation of many tools for parsing and concomitant software that are available on the Proteus website under software. These include several parsers plus COMLEX, a dictionary that contains exceptionally detailed syntactic information, and NOMLEX, a dictionary of English nominalizations that describes the allowed complements for a nominalization and the nominal complements to the arguments of the corresponding verb.

By the spring of 1973, I had also completed my dissertation (On Endpoint Distribution of Self-avoiding Walks) and was offered a faculty position in Computer Science at NYU. This gave me a solid foundation for continuing with Naomi and her Linguistic

[3] This idea was modified for MUC-6, as discussed below.

String Project team. It also gave me the opportunity to engage in a project I had dreamed about since my introduction to computers in 1964: to design and build a high-speed microprogrammed machine, PUMA, to emulate the Control Data 6600 machine at a much lower cost.

6. Sublanguage and "Information Formatting"

As we dealt with extracting information from the medical domain that we were examining—patient discharge summaries—we became interested in the portability of the sublanguage rules we were developing.

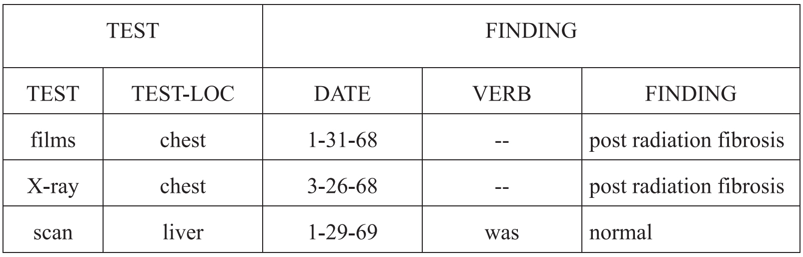
Harris (1968) had proposed a theory of sublanguages that makes them amenable to computation even though they lack conformance with the standard language. The sublanguage is defined by the standard language words that are omitted. It is smaller than the standard language due to "zeroing" (*Work request submitted. Crankshaft scoured.*) but informationally richer due to domain specific co-occurrence patterns in the sublanguage.

For example, in Figure 2, *film of chest (shows) post radiation fibrosis*, *X-ray of chest (shows) post radiation fibrosis*, and *scan of chest was normal* indicate that *film, X-ray,* and *scan* form a subclass of nouns in texts of this subfield based on the similarity of their environments. Such categorization allows us to build up word classes and their relations to their "neighbors" that reflect the specialized use of language in the subdiscipline and allow us to generate numerical summaries of the different types of information using what Sager refers to as an "information format," such as the "Simple radiology format" in Figure 2.

The insights of Harris and Sager were prescient as they came before the rapid growth of text information available in digital form. As Internet access grew, the goal of finding information from unstructured text grew with it in the discipline of IE.

7. Sublanguage Studies at the Navy Research Laboratory

As we started to present our sublanguage results on the medical domain, the Navy Research Lab (NRL) came to us with messages on equipment failures, called casualty reports, or CASREPS. These CASREPS looked interestingly parallel to our patient data. After all, patients suffer physical failures and are treated, while machines suffer mechanical failures and are repaired.



TEST		FINDING		
TEST	TEST-LOC	DATE	VERB	FINDING
films	chest	1-31-68	--	post radiation fibrosis
X-ray	chest	3-26-68	--	post radiation fibrosis
scan	liver	1-29-69	was	normal

Figure 2
Simple radiology format from Sager (1981).

This was followed by an invitation from NRL to work on turning the Navy's CASREPS into usable information on mechanical failures and other ship-board problems. The joint Navy/NYU project identified several related semantic patterns shared by the medical and CASREP data and a shared domain, failures, that showed similar relationships between objects in the two sublanguages (Marsh 1986). The String Project system was also ported to NRL and adapted to the Navy messages (Marsh and Friedman 1985).

8. The DARPA Strategic Computing Program

One notable characteristic of IE is the degree to which its research has been driven by a series of U.S. Government–sponsored evaluations that began in the mid 1980s under the rubric TIPSTER with the goal of engaging in speech and text research and development efforts to support the needs of DARPA, the Defense Advanced Research Projects Agency.

The efforts began in the mid 1980s with DARPA's Speech R&D Program, which was based on an Evaluation Driven Research Paradigm (Prange 1996). The Paradigm saw a clearly defined final objective for the speech work, a set of specific tasks that would move the R&D community significantly closer to the program's final objective—a tailored metric and evaluation methodology to measure progress toward accomplishing each of the chosen tasks, sufficient quantities of training and testing data, and a group of several leading-edge research institutions that would periodically participate in formal evaluations, including open discussion of their successes and failures in completing the assigned tasks. The model, housed in DARPA's National Institute of Standards and Technology (NIST), has continued to this day under several programs with evolving goals. I present this evolution as a history of IE (with dates no less) in which I was heavily involved.

As for text analysis, by the mid 1980s there had been several efforts at information extraction from news (DeJong 1982) and medical reports (Sager et al. 1987), but evaluation was limited, particularly with regard to comparing one system with another, so it was hard to tell whether progress was being made. To address this issue, DARPA advanced the Evaluation Driven Research Paradigm, which had been working successfully with speech, for its fledgling text program, providing a substantial chunk of money in 1984 for NLP, in particular for message understanding and information extraction. It spent the money on competitions among information extraction and message understanding groups at carefully designed "conferences," fondly called MUCs (Message Understanding Conferences). This resulted in a large group of participants who competed in seven MUC "conferences," with advances being made in each competition. Over the MUCs and the following competitions in ACE and KBP, this resulted in constant improvement in each task being evaluated. A particular advantage of the paradigm was that improvements in one task often fed improvements in a related task.

Naomi Sager remained centered on medical applications, with funding from NSF and the Library of Medicine, so there was a need for a new group to be formed at NYU for the DARPA work. We named the group Proteus (PROtotype TExt Understanding System), with a nod to the Greek sea god who answered only to those who were capable of catching him. Between 1986 and 2018, the Proteus Project produced over 300 publications on message understanding and its supporting software. The project has been funded by grants and contracts from DARPA, the National Science Foundation, and the Linguistic Data Consortium.

9. The MUCs[4] (1987–1998)

For each MUC, participating groups[5] were given a training set and instructions on the type of information to be extracted. Then, shortly before the conference, participants received a set of test messages to be run through their system, without making changes to the system. The output of each system was then evaluated against a manually prepared answer key. The evaluations began in 1989 and were conducted by Beth Sundheim from the Naval Ocean Systems Center (NOSC). They were begun with funding directly from DARPA and continued with this funding under the TIPSTER Program. The MUCs were remarkable in that they in large part shaped the research program in information extraction and brought it forward.

The development corpus for MUC-1 consisted of 10 Navy operations report (oprep) messages, each with a few sentences. Participants each proposed a suitable output for each message.

By MUC-2, the task had crystallized as template filling, with information filling slots in the template such as the type of event, the agent, the time and place, the effect, and so forth. For MUC-2 the template had 10 slots and involved sanitized forms of military messages about naval sightings and engagements.[6] All participants in the conference developed software systems that performed natural language understanding tasks defined by the conference committee templates.

MUC-2 also worked out the primary evaluation measures of *recall* and *precision*. The systems were evaluated based on how their output compared with the output of human linguists. For MUC-3 (1991), the task shifted to reports of terrorist events in Central and South America, as reported in articles provided by the Foreign Broadcast Information Service, and the template became somewhat more complex, with 18 slots, as shown in the MUC-3 template in Figure 3.

MUC-4 (1992) used the same task as MUC-3, with a further small increase in template complexity to 24 slots.

One innovation of MUC-5 (1993) was the use of a nested template structure. In earlier MUCs, each event had been represented as a single template, in effect a single record in a database, with a large number of attributes. This format proved awkward when an event had several participants (e.g., several victims of a terrorist attack) and one wanted to record a set of facts about each one. This sort of information could be much more easily recorded in the hierarchical structure introduced for MUC-5, in which there was a single template for an event, which pointed to a list of templates, one for each participant in the event.[7]

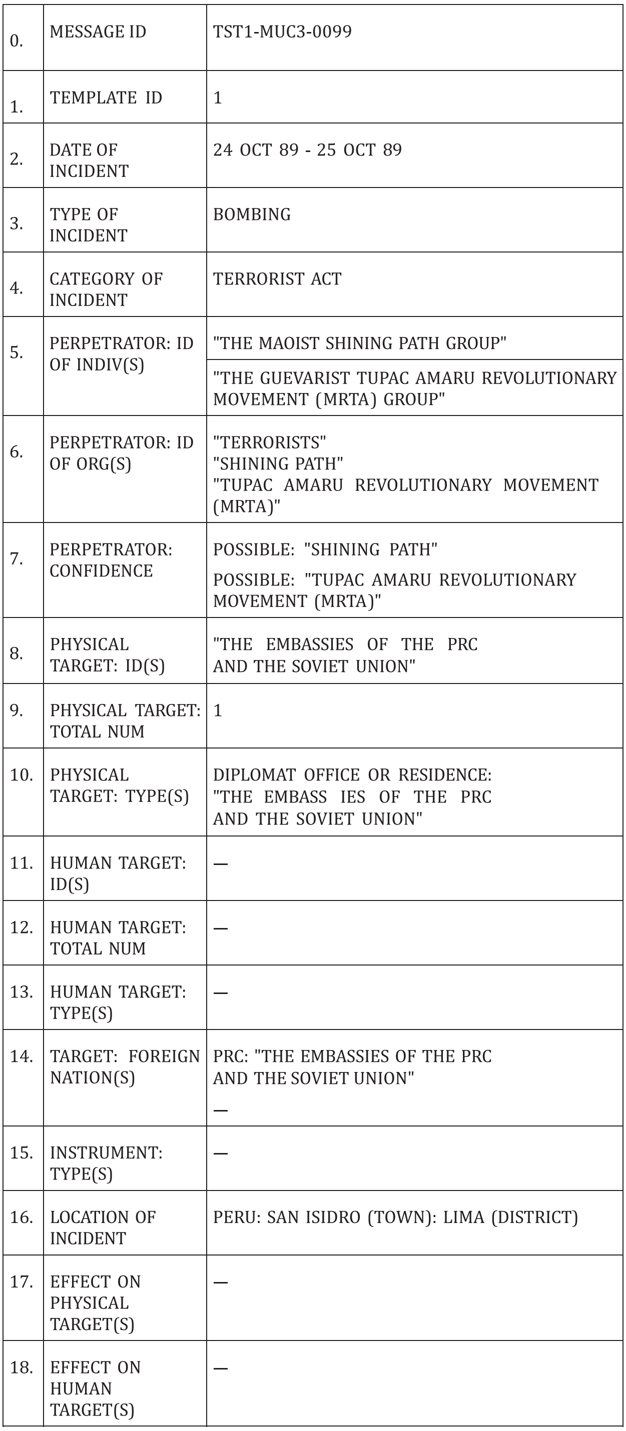
MUC-5 represented a substantial further jump in task complexity. Two tasks were involved, international joint ventures and electronic circuit fabrication, in two languages, English and Japanese. The joint venture task required 11 templates with a total of 47 slots for the output, double the number of slots defined for MUC-4 and the task documentation was over 40 pages long. The MUC-5 tasks were quite complex and a great effort had been invested by the government in preparing the training and test

[4] This section is taken from Grishman, R., & Sundheim, B. M. 1996.

[5] Over the course of the seven conferences, there were 59 sites that participated in one or more evaluations. NIST's Information Extraction site provides a list of the participants.

[6] Proceedings for MUC-2 through MUC-7 are available through the ACL website: `https://aclanthology.org/`.

[7] In fact, the MUC-5 structure was much more complex, because there were separate templates for products, time, activities of organizations, etc.



0.	MESSAGE ID	TST1-MUC3-0099
1.	TEMPLATE ID	1
2.	DATE OF INCIDENT	24 OCT 89 - 25 OCT 89
3.	TYPE OF INCIDENT	BOMBING
4.	CATEGORY OF INCIDENT	TERRORIST ACT
5.	PERPETRATOR: ID OF INDIV(S)	"THE MAOIST SHINING PATH GROUP" "THE GUEVARIST TUPAC AMARU REVOLUTIONARY MOVEMENT (MRTA) GROUP"
6.	PERPETRATOR: ID OF ORG(S)	"TERRORISTS" "SHINING PATH" "TUPAC AMARU REVOLUTIONARY MOVEMENT (MRTA)"
7.	PERPETRATOR: CONFIDENCE	POSSIBLE: "SHINING PATH" POSSIBLE: "TUPAC AMARU REVOLUTIONARY MOVEMENT (MRTA)"
8.	PHYSICAL TARGET: ID(S)	"THE EMBASSIES OF THE PRC AND THE SOVIET UNION"
9.	PHYSICAL TARGET: TOTAL NUM	1
10.	PHYSICAL TARGET: TYPE(S)	DIPLOMAT OFFICE OR RESIDENCE: "THE EMBASS IES OF THE PRC AND THE SOVIET UNION"
11.	HUMAN TARGET: ID(S)	—
12.	HUMAN TARGET: TOTAL NUM	—
13.	HUMAN TARGET: TYPE(S)	—
14.	TARGET: FOREIGN NATION(S)	PRC: "THE EMBASSIES OF THE PRC AND THE SOVIET UNION" —
15.	INSTRUMENT: TYPE(S)	—
16.	LOCATION OF INCIDENT	PERU: SAN ISIDRO (TOWN): LIMA (DISTRICT)
17.	EFFECT ON PHYSICAL TARGET(S)	—
18.	EFFECT ON HUMAN TARGET(S)	—

Figure 3
Template for MUC-3.

data and by the participants in adapting their systems for those tasks. Most participants worked on the tasks for six months, a few (the TIPSTER contractors) had been at work on the tasks for considerably longer.

9.1 MUC-6 and the Classic Information Extraction Tasks

In light of the MUC-5 issues, DARPA convened a meeting of TIPSTER participants and government representatives in December 1993 to define goals and tasks for MUC-6 (Grishman and Sundheim 1996). Among the goals that we identified were:

- Demonstrating task independent component technologies of information extraction that would be immediately useful.
- Encouraging work to make information extraction systems more portable.
- Encouraging work on "deeper understanding."

9.1.1 Immediately Useful Tasks. To meet the first goal, the committee developed the Named Entity task, which basically involves identifying the names of all the people, organizations, and geographic locations in a text. Three other tasks—Coreference Resolution, Relation Extraction, and Event Extraction—evolved from the Portability and Deep Understanding goals.

Named Entity Recognition. In the course of developing systems for the early MUCs, researchers began to see the central role that name identification and classification played in information extraction. Names are very common in text, particularly in news text; furthermore, typically many of the columns in an extracted table are to be filled with names. This led to the introduction of a new, separately evaluated task called named entity recognition (NER). Three classes of names were recognized: persons, organizations, and locations. A more varied set of domains, however, would demand a larger or at least different set of names. Chemistry articles, for example, will contain names of chemicals, whereas biology articles will contain names of species, of proteins, of genes.

To identify names, we used a training corpus annotated for names using standard regular-expression notation, and, given a text, we searched for the most probable consistent sequence of patterns using a Viterbi search. By systematically adding such patterns, it is possible to develop a high-performance name tagger.

Having the probability of a tag depend only on the current word, however, is inadequate. Different name taggers have taken different approaches to combining evidence without requiring enormous amounts of training data. At NYU, a decision-tree name tagger (Sekine et al. 1998), and a Maximum Entropy model (Borthwick et al. 1998) have been constructed by Proteus members. The book by David Nadeau and Satoshi Sekine (2007) provides a survey of NER models.

9.1.2 Portability. The difficulties encountered in MUC-5 led to the goal of portability, the ability to rapidly retarget a system to extract information about a different class of events. The committee felt it was important to demonstrate that useful extraction systems could be created in a few weeks. To meet this goal, we decided that the IE task for MUC-6 would have to involve a relatively simple template, more like MUC-2 than MUC-5; this was dubbed “mini-MUC.” In keeping with the hierarchical template structure introduced in MUC-5, it was envisioned that the mini-MUC would have an event-level template pointing to templates representing the participants in the event (people, organizations, products, etc.) mediated perhaps by a “relational” level template.

The Template Element (Relation Extraction). Identifying the names in the text is not informative without knowing how the named entities relate to the specifications of the extraction task. In the case of a named person, this is usually their job, for example,

EmployeeOf (Jack Smith, U.S. Department of Justice)

and other relations that are predefined based on the task, for example,

LocatedIn (Jack Smith, Washington, D.C.)

JobTitle (Jack Smith, Special Counsel)

This task, although superficially similar to named entities—it is also based on identifying people and organizations—is significantly more difficult. One has to identify descriptions of entities (“a distributor of kumquats”) as well as names. If an entity is mentioned several times, possibly using descriptions or different forms of a name, these need to be identified together; there should be only one template element for each entity in an article. Consequently, the scores were appreciably lower, ranging across most systems from 65% to 75% in recall, and from 75% to 85% in precision. The top-scoring system had 75% recall, 86% precision. Systems did particularly poorly in identifying descriptions; the highest-scoring system had 38% recall and 51% precision for descriptions.

There seemed to be general agreement that having prepared code for template elements in advance did make it easier to port a system to a new scenario in a few weeks. This factor, and the room that exists for improvement in performance, suggest that including this task in future evaluations would be worthwhile.

Scenario (Event) Extraction. Ever on the lookout for additional evaluation measures, the committee decided to make the creation of template elements for all the people and organizations in a text a separate MUC task. Like the named entity task, this was also seen as a potential demonstration of the ability of systems to perform a useful, relatively domain-independent task with near-term extraction technology (although it was recognized as being more difficult than named entity, since it required merging information from several places in the text). The old-style MUC information extraction task, based on a description of a particular class of events (a “scenario”) was called the “scenario template” task, exemplified in the Succession event shown here:

```
<SUCCESSION_EVENT-9402240133-3>:=
    SUCCESSION_ORG: <ORGANIZATION-9402240133-1>
    POST: ‘‘vice chairman, chief strategy
       officer, world-wide’’
    IN_AND_OUT: <IN_AND_OUT-9402240133-5>
    VACANCY_REASON: OTH_UNK
```

```
<IN_AND_OUT-9402240133-5>:=
      IO_PERSON: <PERSDN-9402240133-5>
      NEW_STATUS: IN
      ON_THE_JOB: YES
      OTHER_ORG: <ORGANIZATION-9402240133-8>
      REL_OTHER_ORG: OUTSIDE_ORG
<ORGANIZATION-9402240133-1>
      ORG_NAME: ‘‘McCann’’
      ORG_TYPE: COMPANY
<ORGANIZATION-9402240133-8>
      ORG_NAME: ‘‘J. Walter Thompson’’
      ORG_TYPE: COMPANY
<PERSON-9402240133-5>:=
      PER_NAME: ‘‘Peter Kirn’’
```

9.1.3 NYU: Where's the Syntax?[8] The event (scenario) and relational templates from the mini-MUC, along with NER and coreference resolution, have remained in IE as now classic IE text types.

The portability goal also led the NYU team to abandon the full syntactic analysis followed by a semantic analysis of the syntactic structure that had been the backbone of our work and the source of our success since our association with the String Project. The full analysis was just too slow. In processing the language as a whole, our system was operating with only relatively weak semantic preferences. As a result, the process of building a global syntactic analysis involved a large and relatively unconstrained search space and was consequently quite expensive. In contrast, pattern-matching systems assemble structure "bottom–up" and only in the face of compelling syntactic or semantic evidence in a (nearly) deterministic manner.

Speed was particularly an issue for MUC-6 because of the relatively short time frame (one month for training). With a slow system, which can analyze only a few sentences per minute, it is possible to perform only one or at best two runs per day over the full training corpus, severely limiting debugging.

Our system was designed to attempt to generate a full sentence parse if at all possible. If not, it attempted a parse covering the largest substring of the sentence that it could. This global goal sometimes led to incorrect local choices of analyses; an analyzer that trusted local decisions could, in many cases, have done better.

Having a broad-coverage, linguistically principled grammar meant that relatively few additions were needed when moving to a new scenario. However, when specialized constructs did have to be added, the task was relatively difficult, since these constructs had to be integrated into a large and quite complex grammar.

Ultimately, we concluded that we should "do a MUC" ourselves using the pattern-matching approach in order to better appreciate its strengths and weaknesses. In particular, we carefully studied the FASTUS system of Appelt et al. (1993), who have clearly and eloquently set forth the advantages of this approach. This approach can be viewed as a form of conservative parsing, although the high-level structures that are created are not explicitly syntactic.

We exaggerate, of course, the radicalness of our change. Several components were direct descendants of earlier modules: The dictionary was Comlex Syntax (Grishman,

[8] This section is taken from Grishman (1995).

Macleod, and Meyers 1994); the lexical analyzer (for names, etc.) had been gradually enhanced at least since MUC-3; the concept hierarchy code and reference resolution were essentially unchanged from earlier versions. In addition, our grammatical approach was not entirely abandoned; our noun group patterns were a direct adaptation of the corresponding portion of our grammar, just as Hobbs's patterns were an adaptation from his grammar.[9] And, as we shall see, more of the grammar crept in as our effort progressed. In essence, one could say that our MUC-6 system was built (in late August and early September 1995) by replacing the parser and semantic interpreter of our earlier system by additional sets of finite-state patterns.

9.1.4 Deeper Understanding.[10] Another concern that was noted about the MUCs was that the systems were tending towards relatively shallow understanding techniques (based primarily on local pattern matching), and that not enough work was being done to build up the mechanisms needed for deeper understanding. Therefore, the committee, with strong encouragement from DARPA, included three MUC tasks that were intended to measure aspects of the internal processing of an information extraction or language understanding system. These three tasks, which were collectively called SemEval ("Semantic Evaluation") were coreference resolution, word sense disambiguation, and predicate argument structure. Problems arose with each of these tasks and, given that coreference resolution was the only task that was directly relevant to information extraction, it was retained for the MUC-6 formal evaluation.

Coreference Resolution. We have already mentioned coreference explicitly in resolving "he" as equivalent to the person "Fred Flintstone" in Figure 1 and implicitly in identifying the organizations the "Maoist Shining Path Group" and the "Guevarist Tupac Amaru Revolutionary Movement" as perpetrators in Figure 3. Information extraction gathers information about such discrete *entities* which, in addition to people and organizations, includes vehicles, books, cats, and so forth. The texts contain *mentions* of these entities. These mentions may take the form of

- names ("Jack Smith")
- noun phrases headed by common nouns ("special counsel")
- pronouns ("he")

The various stages of pattern matching produce a *logical form* for the sentence, consisting of a set of entities and a set of events that refer to these entities. These must then be integrated with the entities and events from the prior discourse (prior sentences in the article).

In the NYU system for MUC-6, reference resolution examines each entity and event in logical form and decides whether it is an anaphoric reference to a prior entity or event, or whether it is new and must be added to the discourse representation.[11] If

[9] And, since both these grammars can trace their origins in part to the NYU Linguistic String Grammar, the approaches here are very similar.

[10] This section is taken from Grishman and Sundheim (1996).

[11] In some cases, such as apposition, the anaphoric relation is determined by the syntax. Such cases are detected and marked by the pattern-matching stages, and checked by reference resolution before other tests are made.

the noun phrase has an indefinite determiner or quantifier (e.g., "a," "some," "any," "most") it is assumed to be new information. Otherwise, a search is made through the prior discourse for possible antecedents. An antecedent will be accepted if the class of the anaphor (in our classification hierarchy) is equal to or more general than that of the antecedent, if the anaphor and antecedent match in number, and if the modifiers in the anaphor have corresponding arguments in the antecedent. Special tests are provided for names, since people and companies may be referred to by a subset of their full names; a match on names takes precedence over other criteria.

9.1.5 NYU Overall Performance on MUC-6.[12] As one of the perpetrators of this multitask evaluation, NYU felt obliged to participate in all four tasks. Results on the four tasks are as follows:

Overall System Performance			
Task	Recall	Precision	F-measure (P&R)
Named Entity	86	90	88.19
Coreference	53	62	
Template Element	62	83	71.16
Scenario Template	47	70	56.40

Our relative standing on these tasks[13] for the most part accorded with the effort we invested in the tasks over the last few months.

For Named Entity, our pattern set built on work done for previous MUCs. From mid August to early September we spent several weeks tuning Named Entity annotation, using the Dry Run Test corpus for training, and pushed our performance to 90% recall, 94% precision on that corpus. Our results on the formal test, as could be expected, were a few points lower. There was no shortage of additional patterns to add in order to improve performance but at that point our focus shifted entirely to the Scenario Template task.

For the Scenario Template task, we spent the first week studying the corpus and writing some of the basic code needed for the pattern-matching approach, which we were trying for the first time. The remainder of the time was a steady effort of studying texts, adding patterns, and reviewing outputs. Our first run was made 10 days into the test period; we reached 29% recall one week after the first run and 48% two weeks after the first run; our final run on the training corpus reached 54% recall (curiously, precision hovered close to 70% throughout the development period).

For the final system, we attempted to fill all the slots, but did not address some of the finer details of the task. We did not record "interim" occupants of positions, did not do the time analysis required for ON_THE_JOB (we just used NEW_STATUS), and did not distinguish related from entirely different organizations in the RELOTHER_ORG slot. In general, it seemed to us that—given the limited time—adding more patterns yielded greater benefits than focusing on these details.

NYU did relatively well on the Scenario Template task. We can hardly claim that this was the result of a new and innovative system design, since our goal was to gain experience and insight with a design that others had proven successful. Perhaps it

[12] This section is taken from Grishman (1995).

[13] As MUC participants, we are bound by the participation agreement not to disclose other sites' scores, so no direct comparison can be provided.

was a result of including patterns beyond those found in the formal training. In particular, we

- added syntactic variants (relatives, reduced relatives, passives, etc.) of patterns even if the variants were not themselves observed in the training corpus.
- studied some 1987 Wall Street Journal articles related to promotions (in particular, we searched for the phrase "as president"), and added the constructs found there.

Perhaps we just stayed up late a few more nights than other sites.

We did not do any work specifically for the Coreference and Template Element tasks, although our performance on both these tasks gradually improved as a result of work focused on Scenario Templates.

10. ACE: Automatic Content Extraction (1999–2008)[14]

Each of the MUC evaluations was focused on a single topic. The next series of TIPSTER evaluations, Automatic Content Extraction (ACE), aimed to cover news stories more generally, including politics and international affairs, through a broader range of entities, relations, and events. ACE had roughly annual evaluations from 2000 to 2008. Later versions of ACE included multilingual extraction, with texts in English, Chinese, and Arabic. Although there was a small effort at cross-document analysis, almost all the processing in ACE was on a document-by-document basis; systems were judged separately on their ability to correctly extract the information in each document.

The goals of ACE were to identify the MUC tasks that would be advanced for the remainder of the TIPSTER program—in essence, to anoint them as the classic text types that would enable meaning to be identified in unstructured text so that, instead of simply identifying Jack Smith as a name, the task would be to infer which Jack Smith in light of the context in which the name occurred. There is a wide variation in the complexity of the information structures we can try to extract from texts. At one end of the spectrum are elementary events that may take 2 or 3 primary arguments plus optional time and place modifiers. The ACE 2005 event extraction task included 33 such event types, listed in the table just below. The goal in selecting these types was to identify types of events that occurred frequently over a wide range of news stories. No attempt was made to capture the relations between events.

Event type	Subtypes
Life	Be-born, Marry, Divorce, Injure, Die
Movement	Transport
Transaction	Transfer-ownership, Transfer-money
Business	Start-org, Merge-org, Declare-bankruptcy, End-org
Conflict	Attack, Demonstrate
Personnel	Start-position, End-position, Nominate, Elect
Justice	Arrest-jail, Release-parole, Trial-hearing Charge-indict, Sue, Convict, Sentence, Fine, Execute, Extradite, Acquit, Appeal, Pardon

[14] This section is taken from my Tarragona Lectures (Grishman 2012).

A noteworthy feature of ACE was that it had more data and better data thanks to the high-quality data collection and annotation from the Linguistic Data Consortium (LDC).[15] This resulted in better performance and overall higher quality over a range of measures.

11. Knowledge Base Population (KBP) and the Text Analysis Conferences (TAC) (2009–present)

Although there was a small effort at cross-document analysis in ACE, most systems were judged separately on their ability to correctly extract the information in each document. This is the primary aspect that has shifted with the current series of evaluations, Knowledge Base Population (KBP), a track of the annual Text Analysis Conferences (TACs).[16] KBP's goal is to enable the collection of facts about entities that may be scattered across many text sources in English, Spanish, and Chinese and two primary genres, formal newswire and information web text, in order to populate an existing or emerging knowledge base. As with ACE, the LDC has supplied the training and test data for the ever-changing KBP feature requirements and the knowledge bases used for evaluation.[17]

The early KBP evaluations included two tasks, Entity Linking and Slot Filling. Entity Linking requires the matching of named mentions in the corpus with entities in a Knowledge Base or report NIL if there is no KB entry. Slot Filling involves searching the corpus for information about an entity in the KB.

As a task coordinator for the Slot Filling task, I engaged the Proteus team for the first three years of TAC (2010–2012) on this task (Grishman and Min 2010; Sun et al. 2011; Min et al. 2012). Given that the task prioritizes finding the right information rather than all possible mentions, we aimed for a high score in precision.

For each year, our performance ranked above 50%, with pattern bootstrapping procedure starting with a small seed and multiple iterations contributing most to precision.

More recently, work in KBP has used the properties of large language models to refine the populating of a knowledge base, as exemplified by Chi Han et al. (2024). Han et al. "theoretically and empirically revisit output word embeddings and find that their linear transformations are equivalent to steering language model generation styles," resulting in "comparable or superior performance" on tasks such as language model detoxification and sentiment control compared with controlled generation methods.

12. Current Work in Information Extraction

For all IE applications, performance hinges on the quality and quantity of the training data and on the matching of this data to the text to be classified. The advent of large language models and the ability to search them with context-aware language models like BERT have changed information extraction in promising ways, as many of the papers at this conference demonstrate—with a shout out to our own Heng Ji. It's an exciting time to be in the IE field; I look forward to seeing your results.

[15] For size and the variety of quality checks on the ACE data, see Doddington et al. 2004.

[16] Text Analysis Conference proceedings are available at `http://www.nist.gov/tac/publications/index.html`.

[17] See Getman et al. (2018) for information on data that has addressed the KBP requirements.

References

Appelt, D., J. Hobbs, J. Bear, D. Israel, M. Kameyama, and M. Tyson. 1993. SRI: Description of the JV FASTUS system used for MUC-5. In *Proceedings of the Fifth Message Understanding Conference (MUC-5)*. https://doi.org/10.3115/1072017.1072039

Borthwick, A., J. Sterling, E. Agichtein, and R. Grishman. 1998. Exploiting diverse knowledge sources via maximum entropy in named entity recognition. In *Sixth Workshop on Very Large Corpora*.

DeJong, Gerald. 1982. An overview of the FRUMP system. In Wendy Lehnert and Martin Ringle, editors, *Strategies for Natural Language Processing*, pages 149–176. Lawrence Erlbaum.

Doddington, G. R., A. Mitchell, M. A. Przybocki, L. A. Ramshaw, S. M. Weischedel, and R. M. Strasseland. 2004. The automatic content extraction (ACE) program-tasks, data, and evaluation. *Lrec*, 2(1):837–840.

Getman, J., J. Ellis, S. Strassel, Z. Song, and J. Tracey. 2018. Laying the groundwork for knowledge base population: Nine years of linguistic resources for TAC KBP. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*.

Grishman, Ralph. 1971. *Assembly Language Programming for the Control Data 6000 Series*. Algorithmics Press.

Grishman, Ralph. 1995. The NYU System for MUC-6 or Where's the Syntax? In *Sixth Message Understanding Conference (MUC-6): Proceedings*. https://doi.org/10.21236/ADA460232

Grishman, Ralph. 2003. *The Oxford Handbook of Computational Linguistics*, edited by Ruslan Mitkov, chapter 30. Oxford University Press.

Grishman, Ralph. 2012. Information extraction: Capabilities and challenges. *International Winter School in Language and Speech Technologies WSLST*, volume 41. https://doi.org/10.1093/oxfordhb/9780199276349.013.0030

Grishman, Ralph, Catherine Macleod, and Adam Meyers. 1994. Comlex Syntax: Building a computational lexicon. In *Proceedings of the 15th International Conference on Computational Linguistics (COLING 94)*, pages 268–272. https://doi.org/10.3115/991886.991931

Grishman, Ralph and Bonan Min. 2010. New York University KBP 2010 Slot Filling system. In *Proceedings of Text Analysis Conference (TAC 2010)*.

Grishman, Ralph and Beth Sundheim. 1996. Message Understanding Conference-6: A brief history. In *COLING 1996: The 16th International Conference on Computational Linguistics*, volume 1. https://doi.org/10.3115/992628.992709

Han, Chi, Jialiang Xu, Manling Li, Yi Fung, Chenkai Sun, Han Jiang, Tarek Abdelzaher, and Heng Ji. 2024. Word Embeddings are Steers for Language Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL2024)*. (Outstanding Paper Award).

Harris, Zellig. 1958. Linguistic transformations for information retrieval. In *Proceedings of the International Conference on Scientific Information*.

Harris, Zellig. 1968. *Mathematical Structures of Language*. John Wiley & Sons Inc.

Marsh, Elaine. 1986. General semantic patterns in different sublanguages. In Ralph Grishman and Richard Kittredge, editors, *Analyzing Language in Restricted Domains*. Lawrence Erlbaum Associates, pages 103–128.

Marsh, Elaine and Carol Friedman. 1985. Transporting the Linguistic String Project system from a medical to a Navy domain. *ACM Transactions on Information Systems (TOIS)*, 3(2):121–140. https://doi.org/10.1145/3914.3984

Min, Bonan, Xiang Li, Ralph Grishman, and Ang Sun. 2012. New York University 2012 System for KBP Slot Filling. In *Proceedings of the 2012 Text Analysis Conference*.

Nadeau, David and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26. https://doi.org/10.1075/li.30.1.03nad

Prange, J. D. 1996. Evaluation driven research: The foundation of the TIPSTER text program. In *TIPSTER TEXT PROGRAM PHASE II: Proceedings of a Workshop*, pages 13–22. https://doi.org/10.3115/1119018.1119022

Sager, Naomi. 1981. *Natural Language Information Processing*. Addison-Wesley.

Sager, Naomi, Carol Friedman, Margaret Lyman, and members of the Linguistic String Project. 1987. *Medical Language Processing: Computer Management of Narrative Data*. Addison-Wesley.

Sekine, Satoshi, Ralph Grishman, and H. Shinnou. 1998. A decision tree method for finding and classifying names in Japanese texts. In *Proceedings of the Sixth Workshop on Very Large Corpora*.

Sun, Ang, R. Grishman, W. Xu, and B. Min. 2011. New York University 2011 System for KBP Slot Filling. In *Proceedings of the TAC 2011 Workshop*.