Information Extraction

Jim Cowie and Yorick Wilks

1. Introduction

Information Extraction (IE) is the name given to any process which selectively structures and combines data which is found, explicitly stated or implied, in one or more texts. The final output of the extraction process varies; in every case, however, it can be transformed so as to populate some type of database. Information analysts working long term on specific tasks already carry out information extraction manually with the express goal of database creation.

One reason for interest in IE is its role in evaluating, and comparing, different Natural Language Processing technologies. Unlike other NLP technologies, MT for example, the evaluation process is concrete and can be performed automatically. This, plus the fact that a successful extraction system has immediate applications, has encouraged research funders to support both evaluations of and research into IE. It seems at the moment that this funding will continue and will bring about the existence of working systems. Applications of IE are still scarce. A few well known examples exist and other classified systems may also be in operation. It is certainly not true that the level of the technology is such that it is easy to build systems for new tasks, or that the levels of performance are sufficiently high for use in fully automatic systems. The effect on long term research on NLP is debatable and this is considered in the final section which speculates on future directions in IE.

We begin our examination of IE by considering a specific example from the Fourth Message Understanding Conference (MUC-4 DARPA '92) evaluation. An examination of the prognosis for this relatively new, and as yet unproven, language technology follows together with a brief history of how IE has evolved is given. The related problems of evaluation methodology and task definition are examined. The current methods used for building IE extraction systems are outlined. The term IE can be applied to a range of tasks, and we consider three generic applications.

1. An Example: The MUC-4 Terrorism Task

The task given to participants in the MUC-4 evaluation (1991) was to extract specific information on terrorist incidents from newspaper and newswire texts relating to South America. Human analysts (in this case the participants in the evaluation) prepared training and test data by performing human extraction from a set of texts. The templates to be completed, either by humans, or by computers, consisted of slot labels, and rules as to how the slot was to be filled. For MUC-4 a flat record structure was used, slots which had no information being left empty. Without further commentary we give a short text and its associated template:

SANTIAGO, 10 JAN 90 -- [TEXT] POLICE ARE CARRYING OUT INTENSIVE OPERATIONS IN THE TOWN OF MOLINA IN THE SEVENTH REGION IN SEARCH OF A GANG OF ALLEGED EXTREMISTS WHO COULD BE LINKED TO A RECENTLY DISCOVERED ARSENAL. IT HAS BEEN REPORTED THAT CARABINEROS IN MOLINA RAIDED THE HOUSE OF 25-YEAR-OLD WORKER MARIO MUNOZ PARDO, WHERE THEY FOUND A

FAL RIFLE, AMMUNITION CLIPS FOR VARIOUS WEAPONS, DETONATORS, AND MATERIAL FOR MAKING EXPLOSIVES.

IT SHOULD BE RECALLED THAT A GROUP OF ARMED INDIVIDUALS WEARING SKI MASKS ROBBED A BUSINESSMAN ON A RURAL ROAD NEAR MOLINA ON 7 JANUARY. THE BUSINESSMAN, ENRIQUE ORMAZABAL ORMAZABAL, TRIED TO RESIST; THE MEN SHOT HIM AND LEFT HIM SERIOUSLY WOUNDED. HE WAS LATER HOSPITALIZED IN CURICO. CARABINEROS CARRIED OUT SEVERAL OPERATIONS, INCLUDING THE RAID ON MUNOZ' HOME. THE POLICE ARE CONTINUING TO PATROL THE AREA IN SEARCH OF THE ALLEGED TERRORIST COMMAND.

FIGURE 1. Extracted Terrorism Template

Template Slot ID	Fill Value DEV-MUC3-0017 (NCCOSC)					
0. MESSAGE: ID						
1. MESSAGE: TEMPLATE	1					
2. INCIDENT: DATE	07 JAN 90					
3. INCIDENT: LOCATION	CHILE: MOLINA (CITY)					
4. INCIDENT: TYPE	ROBBERY					
5. INCIDENT: STAGE OF EXECUTION	ACCOMPLISHED					
6. INCIDENT: INSTRUMENT ID	-					
7. INCIDENT: INSTRUMENT TYPE	GUN: "-"					
8. PERP: INCIDENT CATEGORY	TERRORIST ACT					
9. PERP: INDIVIDUAL ID	"ARMED INDIVIDUALS" /					
	"GROUP OF ARMED INDIVIDUALS WEARING SKI MASKS" /					
	"MEN"					
10. PERP: ORGANIZATION ID	-					
11. PERP: ORGANIZATION CONFIDENCE -	-					
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	"ENRIQUE ORMAZABAL ORMAZABAL"					
19. HUM TGT: DESCRIPTION	"BUSINESSMAN": "ENRIQUE ORMAZABAL ORMAZABAL"					
20. HUM TGT: TYPE	CIVILIAN: "ENRIQUE ORMAZABAL ORMAZABAL"					
21. HUM TGT: NUMBER	1: "ENRIQUE ORMAZABAL ORMAZABAL"					
22. HUM TGT: FOREIGN NATION	-					
23. HUM TGT: EFFECT OF INCIDENT	INJURY: "ENRIQUE ORMAZABAL ORMAZABAL"					

The template illustrates the two of the basic types of slot; strings from the text e.g. "ENRIQUE ORMAZABAL ORMAZABAL", and "set fills" in which one of a set of predetermined categories must be selected e.g. ROBBERY, GUN, ACCOMPLISHED. On the surface the problem appears reasonably straightforward. The reader should bear in mind, however, that the definition of a template must be precise enough to allow human analysts to produce consistent filled templates (keys) and also give clear guidelines to the builders of automatic systems. We return to these problems in Section

2. Information Extraction: A core language technology

IE technology has not yet reached the market but it could be of great significance to information end-user industries of all kinds, especially finance companies, banks, publishers and governments. For instance, finance companies want to know facts of the following sort and on a large scale: what company take-overs happened in a given time span; they want widely scattered text information reduced to a simple data base. Lloyds of London need to know of daily ship sinkings throughout the world and pay large numbers of people to locate them in newspapers in a wide range of languages. All these are potential uses for IE.

Computational linguistic techniques and theories are playing a strong role in this emerging technology, which should not be confused with the more mature technology of Information Retrieval (IR), which selects a relevant subset of documents from a larger set. IE extracts information from the actual text of documents. Any application of this technology is usually preceded by an IR phase, which selects a set of documents relevant to some query--normally a string of features or terms that appear in the documents. So, IE is interested in the structure of the texts, whereas one could say that, from an IR point of view, texts are just bags of words.

You can contrast these two ways of envisaging text information and its usefulness by thinking about finding, from the World Wide Web, what TV programs you might want to watch in the next week: there is already a web site in operation with text descriptions of the programs on 25 or more British TV channels, more text than most people can survey easily at a single session. On this web site you can input the channels or genre (e.g. musicals, news etc.) that interest you and the periods when you are free to watch. You can also specify up to twelve words that can help locate programs for you, e.g. stars' or film directors' names. The web site has a conventional IR engine behind it, a standard boolean function of the words and genre/channel names you use. The results are already useful--and currently free--- and treat the program descriptions as no more than "bags of words".

Now suppose you also wanted to know what programs your favorite TV critic liked: and suppose the web site also had access to the texts of recent newspapers. An IR system cannot answer that question because it requires searching review texts for films and seeing which ones are described in favorable terms. Such a task would require IE and some notion of text structure. In fact, such a search for program evaluations is not a best case for IE, and we mention it only because it is an

example of the kind of leisure and entertainment application that will be so important in future informatics developments. To see that one only has to think of the contrast between the designed uses and the actual uses of the French Minitel! system--designed for phone number information but actually used largely as an adult dating service.

Some extraction tasks push out the limits of extracting structured information in a standard form, In fact any task with an evaluative component: e.g. one can search movie reviews for directors and actors--even for films where an individual appears in the non-standard role, such as Mel Gibson as a director, and that is a difficult task for an IR system----those are potentially matchable to templates, but a much harder task is to decide if a movie review is positive or not. It is said that US Congressmen, who receive vast amounts of e-mail that they almost certainly cannot read, would welcome any IE system that could tell them simply, of the content of each e-mail message. The result of such a component could clearly be expressed as a template--what is unclear is how one could fill it in a reliable manner.

An important insight, even after accepting our argument that IE is a new, emergent technology, is that what may seem to be wholly separate information technologies are really not so: MT and IE, for example, are just two ways of producing information to meet people's needs and can be combined in differing ways: for example, one could translate a document and then extract information from the result or vice-versa, which would mean just translating the contents of the resulting templates. Which of these one chose to do might depend on the relative strengths of the translation systems available: a simpler one might only be adequate to translate the contents of templates, and so on. This last observation emphasizes that the product of an IE system--the filled templates-- can be seen either as a compressed, or summarized, text itself, or as a form of data base (with the fillers of the template slots corresponding to conventional database fields). One can then imagine new, learning, techniques like data mining being done as a subsequent stage on the results of IE itself.

3. Information Extraction: A Recent Enthusiasm

Extracting information from text as a demonstration of "understanding" goes back to the early days of NLP. Early work by DeJong ('79) at Yale University was on searching texts with a computer to fill predetermined slots in structures, called scripts by his advisor Schank ('77), but which were close to what would now more usually be called templates: structures with predetermined slots to be filled in specified ways, as a Film Director slot would be filled with a name and a ShipSinkingName slot would be filled with a ship's name. Film evaluations are not very script like, but the scenario of ships sinking (needed by Lloyds of London), or the patterns of company take-overs, are much more template/scenario like and suitable for IE techniques.

Early commercially used systems like JASPER (from Carnegie Group) (Andersen et al '86), built for Reuters depended on very complex hand-crafted templates, made up by analysts and a very specific extraction task. However, the IE movement has grown by exploiting, and joining, the recent trend towards a more empirical and text based computational linguistics, that is to say by putting less emphasis on linguistic theory and trying to derive structures and various levels of linguistic generalization from the large volumes of text data that machines can now manipulate.

Information Extraction, particularly in the context of automatic evaluation against human produced results, is a relatively new phenomenon. The early Message Understanding Conferences, in 1987 and 1989, processed naval ship-to shore messages. A move was then made to extract terrorism information from general newspaper texts. The task of developing the human produced keys (template structures filled with data for specific texts) was shared among the MUC participants themselves. The combination of an evaluation methodology and a task which has definite applicability, and appears practicable, attracted the attention of various U.S. government agencies, who were prepared to pay for the development of large numbers of keys using professional information analysts. IE as a subject and standards of evaluation and success up to MUC-5 were surveyed in (Lehnert & Cowie 1996), and broadly one can say that the field grew very rapidly when ARPA, the US defense agency, funded competing research groups to pursue IE, based initially on scenarios like the MUC-4 terrorism events. To this task were added the domains of joint ventures and micro-electronics fabrication developments, with extraction systems for two languages, English and Japanese. All these tasks represent domains where the funders want to replace the government analysts who read the newspapers and then fill templates: when and where, a terrorist event took place, how many casualties etc. Automating this painful human activity is the goal of IE.

A fairly stable R&D community has arisen around the Message Understanding Conferences. As well as the U.S. participants, a few groups from Europe, Canada, and Japan have also been involved. The idea of a common task as a stimulus to research is a useful one, but it also has dangers. In particular, getting so focused on performing well in the evaluation may actually force people to follow avenues which are only short term solutions. The other drawback is that the amount of software development needed to produce the specific requirements of an extraction system are very large. A common plea at the MUC organizing committee is "let's not have the next one next year, that way we'll get some time to do research". On the other hand some actually usable technologies are appearing as a result of the focus on IE and the visibility of the evaluations to funders, both government and commercial. Recognizing and classifying names in text, not a task of particular interest to the NLP community, now proves to be possible at high levels of accuracy. That IE provides a good focus for NLP research is debatable. One key requirement for making IE a usable technology is developing the ability to produce IE systems rapidly without using the full resources of an NLP research laboratory. The most recent MUCs have introduced a task, "co-reference evaluation", with the goal of stimulating more fundamental NLP research.

The trend inside the ARPA Tipster Text Initiative, which provides funding for research on IE and IR, is to attempt to standardize NLP around a common architecture for annotating documents (Grishman 96). This has proved useful for building multi-component NLP systems which share this common representation. CRL's Temple machine translation system (Zajac 96), and Oleada language training system (Ogden 96) both use this representation system as does the Sheffield GATE system described later in this article. This quest for some kind of standardization is now extending to specifying the kind of information (patterns basically) which drive IE systems. More formally the idea is to have a common representation language that different developers can use to share pieces of an extraction system. Thus if someone has expended a lot of effort recognizing information on people in texts this can be incorporated into someone else's system to recognize changes in holders of particular jobs. Re-usable components

of this type would certainly reduce the duplication of effort which is currently occurring in the MUC evaluations.

4. Evaluation and Template Design

Evaluation is carried out for IE by comparing the templates produced automatically by an extraction program with templates for the same texts produced by humans. The evaluation can be fully automatic. Thus analysts produce a set of filled out templates or keys using a computer tool to ensure correct formatting and selection of fields. The automatic system produces its templates in the same form and a scoring program then produces sets of results (see Figure 1, "Overview of the Development of an Extraction System," on page 6 below) for every slot

Most of the MUC evaluations have been based on giving a one point score for every slot correctly filled (Correct). Spurious slots (S) are also counted, these are slots that are generated, and filled, despite there being no information in the text, and slots with incorrect fills (I). The total number of correct slots (TC) in a template (or key) is also known. These numbers allow two basic scores to be calculated; PRECISION, a measure of the percentage correctness of the information produced, and RECALL, a measure of the percentage of information available which is actually found.

These measures are adapted from information retrieval, and are not so appropriate for IE. For example in an object-style template, if a pointer to a person slot is filled, this counts as a correct fill, then if the name is filled in the person object, this counts as a second correct fill. Another problem comes when there are multiple objects filling a slot how should the scoring system match these up with the multiple objects generated by the human analyst? For example an object may contain a company name and a location. If two of these objects are created one with an empty name slot and the location slot filled, and the other with data in the company name slot and in the location slot. There are also two objects in the key if they are aligned in one order the company slot is correct, but both locations are incorrect. Aligned in the opposite order the company name is incorrect, but both locations are correct. The method of counting correct slots can produce some paradoxical results for a system's scores. In MUC3 one single key which was filled with information about the killing of Jesuit priests was used as the extracted information for each of the test documents. This gave scores as good as many systems which were genuinely trying to extract information. Similarly a set of keys with only pointers to objects and no strings in any other slots was submitted by George Krupka (GE at that time) in MUC-5. This scored above the median of system performance. The point really is that the details of how a score were achieved is important. A 100% recall IR system is easy to build too, just retrieve all the documents.

FIGURE 2. A Partial View of System Summary Scores - Micro-Electronics Template

SLOT	POS	ACT	COR	PAR	INC	SPU	MIS	REC	PRE
<template></template>	100	100	100	0	0	0	0	100	100
content	123	134	94	0	2	38	27	76	70
subtotals	123	134	94	0	2	38	27	76	70
<entity></entity>	121	131	91	0	9	31	21	75	69
name	121	131	77	3	20	31	21	65	60
location	58	47	25	4	4	14	25	46	57
nationality	36	19	14	0	4	1	18	39	74
type	121	131	91	0	9	31	21	75	69
subtotals	336	328	207	7	37	77	85	63	64
<micro-process></micro-process>	124	134	94	0	2	38	28	76	70
process	124	134	84	0	12	38	28	68	63
developer	63	91	23	0	9	59	31	36	25
manufacturer	83	141	43	0	15	83	25	52	30
distributor	80	138	45	0	9	84	26	56	33
purchaser	25	36	13	0	1	22	11	52	36
subtotals	375	540	208	0	46	286	121	55	38
<layering></layering>	44	57	36	0	1	20	7	82	63
type	44	57	32	2	3	20	7	75	58
film	13	2	0	0	1	1	12	0	0
temperature	5	5	2	0	0	3	3	40	40
device	13	9	6	0	0	3	7	46	67
equipment	39	57	20	0	13	24	6	51	35
subtotals	114	130	60	2	17	51	35	54	47
lithography>	51	47	35	0	1	11	15	69	74
subtotals	161	165	90	5	12	58	54	57	56
<etching></etching>	17	15	9	0	1	5	7	53	60
subtotals	39	37	16	2	5	14	16	44	46
<packaging></packaging>	12	15	10	0	0	5	2	83	67
subtotals	35	40	25	0	0	15	10	71	62

To give a flavor of what an IE system developer faces during an evaluation we present a much reduced set of summary scores for the MUC-5 "micro-electronics" task (Figure 2, "A Partial View of System Summary Scores - Micro-Electronics Template," on page 8). This presents total scores for a batch of documents. Individual scores by document are also produced by the scoring program. The first column shows the names of the slots in the template objects. New objects are marked by delimiting angle brackets. The next columns are; the number of correct fills for the slot, the number of fills produced by the system, the number correct, the number partially matched (i.e. part, but not all, of a noun phrase recognized), the number incorrect, the number generated which have no equivalent in the human produced key ("SPU"rious), the number missing, and finally the recall and precision scores for this slot. At the end of the report are scores total scores for all slots, total scores for only slots in matched objects, a line showing how many texts had templates correctly generated. Finally a score is given, the F measure, which combines retrieval and precision into one number. This can be weighted to favor recall or precision (P&R, 2P&R,P&2R).

The Effects of Evaluation

The aim of evaluation is to highlight differences between NLP methods; to show improvement in the technology over time, and to push research in certain directions. These goals are somewhat in conflict as we will now show. One result of the whole evaluation process has been to push most of the successful groups into very similar methods based on finite state automata and partial parsing (Appelt 95, Grishman 96a). One key factor in the development process is to be able to score test runs rapidly and evaluate changes to the system. Slower more complex systems are not well suited for this rapid test development cycle. The demonstration of improvement over time implies that the same tasks be attempted repeatedly, year, after year. This is an extremely boring prospect for system developers and the MUC evaluations have moved to new tasks in every other evaluation. Making a comparison of performance between old and new tasks is extremely difficult.

The whole scoring system, coupled with the public evaluation process, can actually result in decisions being made in system development which are incorrect in terms of language processing, but which happen to give better scores.

One novel focus produced by IE is in what Donald Walker once called the "Ecology of Language". Most NLP research was concerned with problems which were most easily tested with sentences containing no proper nouns. Why bother then with the idiosyncratic complexities of proper nouns? Walker observed that this "ecology" would have to be addressed before realistic text processing on general text could be undertaken. The IE evaluations have forced people to address this issue and as a result highly accurate name recognition and classification systems have been developed. A separate Tipster evaluation was in fact set up to find if accurate name recognition technology (better than 90% precision and recall) could be produced for languages other than English. The "Multilingual Named Entity Task" (MET) (Merchant 96) was set up in a very short period of time and showed that scores of between 80 and 90% precision and recall were easily achievable for Spanish, Chinese, and Japanese. Markup here was carried out using SGML.

Template Definition

The evaluation methodology depends on a detailed task specification. Without a clear specification the training and test keys produced by the human analysts have low consistency. Often there is a cycle of discovery with new areas of divergence between template designers and human template fillers regularly having to be resolved. This task involves complex decisions, which can have serious implications for the builders of extraction systems.

Defining templates is a difficult task involving the selection of the information elements required, and the definition of their relationships. This applied task has been further complicated in the evaluations by the attempt to define slots which provided "NLP challenges". For example determining if a contract is "being planned", "under execution", or "has terminated". Often these slots became very low priority for the extraction system builders as an attempt to fill them often had seriously prejudicial effects on the system score. Often the best approach is to simply select the most common option.

The actual structure of the templates used has varied from the flat record structure of MUC-4 to a more complex object oriented definition which was used for Tipster and MUC-5 and MUC-6. This groups related information into a single object. For example a person object might contain three strings; name, title, age, and an employer slot, which is a pointer to an organization object. The information contained in both types of representation is equivalent. The newer object style templates make it easier to handle multiple entities which share one slot, as it groups together the information related to each entity in the corresponding object. The readability of the key in printed form suffers as much of it consists of pointers.

The definition consists of two parts; a syntactic description of the structure of the template (often given in a standard form known as BNF - Backus Naur Form), and a written description of how to determine whether a template should be filled and detailed instructions on determining the content of the slots. The description of the Tipster "joint venture" task extended to more than 40 pages. For the simple task of name recognition described below seven pages are used. To see that this detail is necessary consider the following short extract -

4.1 Expressions Involving Elision

Multi-name expressions containing conjoined modifiers (with elision of the head of one conjunct) should be marked up as separate expressions.

"North and South America" <ENAMEX TYPE="LOCATION">North</ENAMEX> and <ENAMEX TYPE="LOCATION">South America</ENAMEX>

A similar case involving elision with number expressions:

"10- and 20-dollar bills" <NUMEX TYPE="MONEY">10</NUMEX>- and <NUMEX TYPE="MONEY">20-dollar</NUMEX> bills

In contrast, there is no elision in the case of single-name expressions containing conjoined modifiers; such expressions should be marked up as a single expression.

"U.S. Fish and Wildlife Service" <ENAMEX TYPE="ORGANIZATION">U.S. Fish and Wildlife Service</ENAMEX>

```
The subparts of range expressions should be marked up as separate expressions.

"175 to 180 million Canadian dollars" <NUMEX TYPE="MONEY">175

</NUMEX> to <NUMEX TYPE="MONEY">180 million Canadian dollars<//i>
// NUMEX>

"the 1986-87 academic year" the <TIMEX TYPE="DATE">1986</TIMEX>-

<TIMEX TYPE="DATE" ALT="87">87 academic year</TIMEX>
```

A short sample of the BNF for the "micro-electronics" task is given below. It should be noted that while the texts provided for this task included many on the packaging of micro-chips they also included a few on the packaging of potato chips.

```
<MICROELECTRONICS CAPABILITY> :=
      PROCESS:
                         (<LAYERING> | <LITHOGRAPHY> | <ETCHING> |
      <PACKAGING>) +
                   DEVELOPER:
                                      <ENTITY> *
                   MANUFACTURER:
                                             <ENTITY> *
                                      <ENTITY> *
                   DISTRIBUTOR:
                   PURCHASER OR USER:
                                             <ENTITY> *
                   COMMENT:
<ENTITY> :=
      NAME:
                         [ENTITY NAME]
                                [LOCATION] *
            LOCATION:
                   NATIONALITY:
                                      [LOCATION COUNTRY ONLY] *
                                      {COMPANY, PERSON, GOVERNMENT,
                   TYPE:
      OTHER}
                   COMMENT:
                                      ٠٠ ٠٠
<PACKAGING> :=
            TYPE:
                                {{PACK TYPE}} ^
                   PITCH:
                                      [NUMBER]
                   PITCH UNITS: {MIL, IN, MM}
                   PACKAGE MATERIAL: {CERAMIC, PLASTIC, EPOXY, GLASS,
                         CERAMIC GLASS, OTHER} *
                   P L COUNT:
                                      [NUMBER] *
                   UNITS PER PACKAGE: [NUMBER] *
                   BONDING:
                                      {{BOND TYPES}} *
                   DEVICE:
                                            <DEVICE> *
                   EQUIPMENT:
                                      <EQUIPMENT> *
                   COMMENT:
```

New Types of Task

Three qualitatively different tasks are now being evaluated at the Message Understanding Conferences-

- Name recognition and classification; (see above)
- Template element creation simple structures linking information on one particular entity;
- Scenario template creation more complex structures linking template elements.

The first two tasks are intended to be domain independent, and the third domain specific. The degree of difficulty ranges from easy for the first through most difficult for the last. It was intended that each task would provide a base support for its successors, however, the un-decomposed output required for the names task may not provide sufficient information to support the template element creation task. The scenario template creation task is distinguished by the fact that a time constraint is placed on the system developers. The specifics of the task are announced a mere month before the evaluation. Thus groups possessing systems which can be rapidly adapted should do well at this task. Groups possessing people with insomnia may also do relatively well!

5. Methods and Tools

Practically every known NLP technique has been applied to this problem. Currently the most successful systems use a finite state automata approach, with patterns being derived from training data and corpora, or specified by computational linguists. The simplicity of this type of system design allows rapid testing of patterns using feedback from the scoring system. The experience of the system developers in linguistics, and in the development of IE systems remains, however, an important factor.

When IE has been attempted for languages other than English the problem appears to be no more difficult. In fact for the Japanese tasks in MUC and Tipster it seemed that the structure of the texts made IE actually easier.

Systems relying solely on training templates and heuristic methods of combination have been attempted with some success for the micro-electronics domain. This means the system is using no NLP whatsoever. In the earlier MUC evaluations there was a definite bias against using the training data to help build the system. By MUC-3 groups were using the training data to find patterns in the text, and to extract lists of organizations and locations. This approach, although successful, has the drawback that only in the early MUC evaluations were hundreds of training keys available.

Much work on learning and statistical methods have been applied to the IE task. This has given rise to a number of independent components which can be applied to the IE task. A conspicuous success has been part-of-speech taggers, systems that assign one and only one part- of-speech symbol (like Proper noun, or Auxiliary verb) to a word in a running text and do so on the basis (usually) of statistical generalizations across very large bodies of text. Recent research (Church 96) has shown that a number of quite independent modules of analysis of this kind can be built up independently from data, usually very large electronic texts, rather than coming from either intuition or some dependence on other parts of a linguistic theory.

These independent modules, each with reasonably high levels of performance in blind tests, include part-of-speech tagging, aligning texts sentence-by-sentence in different languages, syntax analysis, attaching word sense tags to words in texts to disambiguate them in context and so on. That these tasks can be done relatively independently is very surprising to those who believed them all contextually dependent sub-tasks within a larger theory. These modules have been combined in various ways to perform tasks like IE as well as more traditional ones like machine translation (MT). The modules can each be evaluated separately --but they are not in the end real human tasks that people actually do, as MT and IE are.

One can call the former "intermediate" tasks and the latter real or final tasks---and it is really only the latter that can be firmly evaluated against human needs -----by people who know what a translation, say, is and what it is for. The intermediate tasks are evaluated internally to improve performance but are only, in the end, stages on the way to some larger goal. Moreover, it is not possible to have quite the same level of confidence in them since what is, or is not, a correct syntactic structure for a sentence is clearly more dependent on one's commitments to a linguistic theory of some sort, and such matters are in constant dispute. What constitutes proper extraction of people's names from texts, or a translation of it, can be assessed more consistently by many people with no such subjective commitments.

The empirical movement, basing, as it does, linguistic claims on text data, has another stream: the use in language processing of large language dictionaries (of single languages and bilingual forms) that became available about ten years ago in electronic forms from publishers' tapes. These are not textual data in quite the sense above, since they are large sets of intuitions about meaning set out by teams of lexicographers or dictionary makers. Sometimes they are actually wrong, but they have nevertheless proved a useful resource for language processing by computer, and lexicons derived from them have played a role in actual working MT and IE systems (Cowie et al 93).

What such lexicons lack is a dynamic view of a language; they are inevitably fossilized intuitions. To use a well known example: dictionaries of English normally tell you that the first, or main, sense of "television" is as a technology or a TV set, although it is mainly used now to mean the medium itself. Modern texts are thus out of step with dictionaries--even modern ones. It is this kind of evidence that shows that, for tasks like IE, lexicons must be adapted or "tuned" to the texts being analyzed which has led to a new, more creative wave, in IE research: the need not just to use large textual and lexical resources, but to adapt them as automatically as possible, to enable systems to support new domains and corpora. This means both dealing with their obsolescent vocabulary and extending the lexicon with the specialized vocabulary of the new domain.

6. Assembling a Generic IE system

IE's brief history is tightly tied to the recent advances in empirical NLP, in particular to the development and evaluation of relatively independent modules for a range of linguistic tasks, many of which had been traditionally seen as inseparable, or only achievable within some general knowledge-based AI program. It has been something of a surprise to many that such striking results have been achieved in tasks as various as word sense tagging, syntactic parsing,

word sense tagging, sentence alignment between parallel corpora, and so on. "Striking" here means over 95% accuracy, and those who do not find this striking should remember the many years of argument in linguistics and AI that such tasks, however apparently low-level, could not be performed without access to strong theories or knowledge representations.

All this is of strong relevance to IE and to the question of which of such modules, if any, an IE system should consist of, since it now hard to conceive of IE except as some combination of such modules, usually within an overall management "architecture" such as GATE. Hobbs has argued (Hobbs 95) that most IE systems will draw their modules from a fairly predictable set and has specified a "Generic IE System" that anyone can construct like a tinkertoy from an inventory of the relevant modules, cascaded in an appropriate manner. The original purpose of this description was to allow very brief system presentations at the MUC conferences to highlight Most systems contain most of the functionalities their differences from the generic system. described below, but where exactly they occur and how they are linked together varies immensely. Many systems, at least in the early days were fairly monolithic Lisp programs. External forces, such as a requirement for speed, which has meant re-implementation in C or C++, the necessary re-use of external components, such as Japanese segementors, and the desire to have stand-alone modules for proper name recognition, which is reaching the status of a useful commercial product, have imposed new modularity on the IE system. We will retain Hobbs' division of the generic system for a brief exploration of the functionalities required for an IE system. Hobbs' system consists of ten modules:

- 1. a Text Zoner, which turns a text into a set of segments.
- 2. a Preprocessor which turns a text or text segment into a sequence of sentences, each of which being a sequence of lexical items.
- 3. a Filter, which turns a sequence of sentences into a smaller set of sentences by filtering out irrelevant ones.
- 4. a Preparser, which takes a sequence of lexical items and tries to identify reliably determinable small-scale structures.
- 5. a Parser, which takes a set of lexical items (words and phrases) and outputs a set of parse-tree fragments, which may or may not be complete.
- 6. a Fragment Combiner, which attempts to combine parse-tree or logical-form fragments into a structure of the same type for the whole sentence.
- 7. a Semantic Interpreter, which generates semantic structures or logical forms from parse-tree fragments.
- 8. a Lexical Disambiguator, which indexes lexical items to one and only one lexical sense, or can be viewed as reducing the ambiguity of the predicates in the logical form fragments.
- 9. a Coreference Resolver which identifies different descriptions of the same entity in different parts of a text.
- 10. a Template Generator which fills the IE templates from the semantic structures.

We consider in some more detail the functionality of each of these components.

Text Zoner

The zoner uses whatever format information is available from markup information and text layout to select those parts of a text which will actually go through the remainder of the processes. It isolates the rest of the system from the differences in possible text formats. Markup languages such as HTML and SGML (Goldfarb 90) provide the most explicit and well defined structure. Most newswires too support some sort of convention for indicating fielded data in a text. Special fields such as a dateline, giving the location and date of an article, can be recognized and stored in separate internal field. Problematic portions of a text, such as headlines using uppercase, can be isolated for separate treatment. If paragraph boundaries, or tables are also flagged then the zoner is the place to recognize them.

Preprocessor

Sentences are not normally marked, even in SGML documents, so special techniques are required to recognize sentence boundaries. For most languages the main problem here is distinguishing the use of the full stop as a sentence terminator from its use as an abbreviation marker ("Dr., Mr., etc.") and also other idiosyncratic uses (e.g. "..."). Paradoxically languages which appear to be more difficult as they don't use spaces to separate lexical units (Japanese, Chinese) do not have this stop ambiguity problem and can have sentences identified relatively easily.

Once the sentences are identified, or as a part of this process, it is necessary to identify lexical items and possibly convert them to an appropriate form for lexical lookup. Thus, we may convert each word in an English text to uppercase, while still retaining information about its original case usage. Recognizing words is relatively easy in most languages due to the use of spaces. Japanese and Chinese now provide problems and a special purpose segmentation program is normally used to identify (with the ever popular 90% accuracy). The Juman program produced at Kyoto University has been used by many sites as their preprocessor for Japanese. Juman also provides part of speech information and typically this type of lexical information is extracted at this stage of processing.

Filter

The filter process can serve several purposes. Following our argument that IE and IR are natural partners we would normally assume that texts processed by IE have already come through the implicit filter of the IR system. Therefore the assumption is they do contain appropriate information. Many do, but a side effect of the retrieval process is to supply some bogus articles which may pass through an IE system producing incorrect data. A popular example from the microelectronics domain in Tipster was some article on "packaging potato chips" these were not about "packaging micro-electronic chips", the actual topic for the IE system. Thus a filter may attempt to block these texts which are artifacts of the IR process.

The main objective of a filter is to reduce the load on the rest of the system. If relevant paragraphs can be identified then the others can be abandoned. This is particularly important for systems which do extensive parsing on every sentence. The risk here is that paragraphs containing relevant information may be lost.

Normally a filter process will rely on identifying either supporting vocabulary, or patterns, to support its operation. This may be in the inform of simple word counting or more elaborate statistical processing.

Preparser

This stage handles the "ecology of natural language" described earlier and contains what is arguably the most successful of the results of IE so far; proper name identification and classification. Typically numbers (in text or numeric form), dates, and other regularly formed constructions are also recognized here. This may involve the use of case information, special lexicons, and context free patterns, which can be processed rapidly. Often a second pass may be required to confirm shortened forms of names which can not be reliably identified by the patterns, but which can be flagged more reliably once fuller forms of the names are identified. Truly accurate name classification may require some examination of context and usage. Although not common it is possible to provide many instances where simple methods will fail:

- Tuesday Morning a chain of US stores
- Ms. Washington a government staffer
- nCube a company which does not follow normal capitalization conventions
- China a town in Mexico

A sophisticated system will pass all possible options to the next stages of processing, possibly increasing their complexity.

Parser

Most systems perform some type of partial parsing. The necessity of processing actual newspaper sentences, which are often long and very complex, means the development of a complete grammar is impossible. Accurate identification of the structure of noun phrases, and subordinate clauses is, however, possible. This stage may be combined with the process of semantic interpretation described below.

Fragment Combiner

The fragment combiner attempts to produce a complete structure for a sentence. It operates on the components identified in the parser and will use a variety of heuristics to produce a relationship between the fragments. Up to this point it can be argued that the process is domain independent.

Semantic Interpreter

A mapping from the syntactic structures to semantic structures related to the templates to be filled has to be carried out. Systems relate the structures found in a sentence to a specific template using semantic information. Semantic processing may use verb subcategorization information to check if appropriate types are found in the context around a verb or noun phrase. Simple techniques like identifying the semantic types of an apposition may be used to produce certain structures. For example "Jim Smith (human name), chairman (occupation) of XYZ Corp

(Company name)" can produce two template objects; a person, employed by a company, and a company, which has an employee. The imposition of semantic restrictions also produces a disambiguation effect as if inappropriate fillers are found the template elements may not be produced.

At the end of this stage structures will be available which contain fillers for some of the slots in a template.

Lexical Disambiguator

This process can occur either as a side effect of other processes for example the semantic interpreter, or an even earlier filtering stage. It can also be a stand-alone stage prior even to parsing. The process does have to occur somewhere in the system.

Coreference Resolver

Coreference is an important component in further combining fragments to produce fewer, but more completely filled templates. It can be carried out both in the early stages, when pronouns and noun phrases can be linked to proper names using a variety of cues, both syntactic and semantic. It can also be delayed to the final stages when semantic structures can be merged. Both identity, meronymy (part-of relationships), and event coreference are required by an IE system. Reference to the original text, as well as to the semantic structures may be required for successful processing. Strong merging may have the unfortunate effect of merging distinct events as one. Just as unfortunate is the lack of merging identifying two events when in fact only one occurred.

Template Generator

Finally the semantic structures have to be unwound into a structure which can be evaluated automatically or fed into a data-base system. This stage is fairly automatic, but may actually absorb a significant degree of effort to ensure that the correct formats are produced and that the strings from the original text are used.

The System as a Whole

Hobbs is surely right that most or all of these functions will be found somewhere in an IE system: the last by definition. However, the description we give of module 8 shows that it is a process that can be performed early, on lexical items, or later, on semantic structures. There is a great deal of dispute about what appears under module number 5, since some systems use a form of syntactic parser but the majority now prefer some form of direct application of corpus-derived finite-state patterns to the lexical sequences, which is a process that would once have been called "semantic parsing" (Cowie 93).

Other such contrasts could be drawn, and it is probably not correct to assert as some do that, because there is undoubtedly much genericness in IE, there is only one standard IE system and everyone uses it. Another important point to make is that IE is not, as many persist in believing, wholly superficial and theory-free, with no consequences for broader NLP and CL, no matter what the level of success achieved, by the whole technology or by individual modules. many of

the major modules encapsulate highly traditional CL/NLP tasks and preoccupations (e.g. syntactic parsing, word-sense disambiguation, coreference resolution etc.) and their optimization, individually or in combination, to very high levels of accuracy, by whatever heuristics, is a substantial success for the traditional concerns of the field.

7. The Sheffield GATE System

The system designed at the University of Sheffield has been evaluated in two MUCs and it has done particularly well at the named entity task. It incorporates aspects of the earlier NMSU DIDEROT TIPSTER system (Cowie et al '93), and the POETIC system (Mellish et al '92) from the University of Sussex, since some members of those teams joined forces to do IE at Sheffield. There are two aspects of the Sheffield system: first, a software environment called GATE -- General Architecture for Text Engineering (Cunningham '95) -- which attempts to meet the following objectives:

- support information interchange between LE modules at the highest common level possible without prescribing a theoretical approach (though it allows modules which share theoretical presuppositions to pass data in a mutually accepted common form);
- support the integration of modules written in any source language, available either in source or binary form, and be available on any common platform;
- support the evaluation and refinement of LE component modules, and of systems built from them, via a uniform, easy-to-use graphical interface which in addition offers facilities for managing test corpora and ancillary linguistic resources.

GATE owes a great deal to collaboration with the TIPSTER architecture. Secondly, they have built VIE (a vanilla Extraction System) within GATE, one version of which (LaSie) has entered two MUC evaluations (Gaizauskas 95).

GATE Design

GATE comprises three principal elements: GDM, the GATE Document Manager, based on the TIPSTER document manager; CREOLE, a Collection of REusable Objects for Language Engineering: a set of LE modules integrated with the system; and GGI, the GATE Graphical Interface, a development tool for LE R&D, providing integrated access to the services of the other components and adding visualization and debugging tools.

Working with GATE the researcher will from the outset reuse existing components, and the common APIs of GDM and CREOLE mean only one integration mechanism must be learned. And as CREOLE expands, more and more modules will be available from external sources.

VIE: An Application In GATE

Focussing on IE within the context of the ARPA Message Understanding Conferences, has meant fully implementing a system that:

- processes unrestricted `real world' text containing large numbers of proper names, idiosyncratic punctuation, idioms, etc.;
- processes relatively large volumes of text in a reasonable time;
- needs to achieve only a relatively shallow level of understanding in a predefined domain area;
- can be ported to a new domain area relatively rapidly (a few weeks at most).

Given these features of the IE task, many developers of IE systems have opted for robust, shallow processing approaches which do not employ a general framework for 'knowledge representation', as that term is generally understood. That is, there may be no attempt to build a meaning representation of the overall text, nor to represent and use world and domain knowledge in a general way to help in resolving ambiguities of attachment, word sense, quantifier scope, coreference, and so on. Such shallow approaches typically rely on collecting large numbers of lexically triggered patterns for partially filling templates, as well as domain-specific heuristics for merging partially filled templates to yield a final, maximally filled template. This approach is exemplified in systems such as the SRI FASTUS system (Appelt '95) and the SRA and MITRE MUC-6 systems (Kru95,Abe95).

However, this is not the approach that we have taken in VIE. While still not attempting `full' understanding (whatever that might mean), we do attempt to derive a richer meaning representation of the text than do many IE systems, a representation that goes beyond the template itself. Our approach is motivated by the belief, which may be controverted if shallower approaches prove consistently more successful, that high levels of precision in the IE task simply will not be achieved without attempting a deeper understanding of at least parts of the text. Such an understanding requires, given current theories of natural language understanding, both the translation of the individual sentences of the text into an initial, canonical meaning representation formalism and also the availability of general and domain specific world knowledge together with a reasoning mechanism that allows this knowledge to be used to resolve ambiguities in the initial text representation and to derive information implicit in the text.

The key difference between the VIE approach and shallower approaches to IE is that the discourse model and intermediate representations used to derive it in VIE are less task- and template-specific than those used in other approaches. However, while committed to deriving richer representations than many IE systems, we are still attempting to achieve only limited, domain-dependent understanding, and hence the representations and mechanisms adopted still miss much meaning. The approach we have adopted to KR does, nevertheless, allow us to address in a general way the problems of presupposition, co-reference resolution, robust parsing and inference-driven derivation of template fills. Results from the MUC-6 evaluation show that such an approach does no worse overall than shallower approaches and we believe that its generality will, in the long run, lead to the significantly higher levels of precision which will be needed to make IE a genuinely usable NL technology. Meanwhile, we are developing within

_

¹Well, the *Wall Street Journal*.

GATE and using many LaSIe modules, a simpler finite state pattern matcher of the, now, classic type. We will then, within GATE, be able to compare the performances of the two sets of modules.

8. The Future

If we think along these lines we see that the first distinction of this paper, between traditional IR and the newer IE, is not totally clear everywhere but can itself become a question of degree. Suppose parsing systems that produce syntactic and logical representations were so good, as some now believe, that they could process huge corpora in an acceptably short time. One can then think of the traditional task of computer question answering in two quite different ways. The old way was to translate a question into a formalized language like SQL and use it to retrieve information from a database- as in "Tell me all the IBM executives over 40 earning under \$50K a year". But with a full parser of large corpora one could now imagine transforming in the query to form an IE template and searching the WHOLE TEXT (not a data base) for all examples of such employees---both methods should produce exactly the same result starting from different information sources --- a text versus a formalized database.

What we have called an IE template can now be seen as a kind of frozen query that one can reuse many times on a corpus and is therefore only important when one wants stereotypical, repetitive, information back rather than the answer to one-off questions.

"Tell me the height of Everest?", as a question addressed to a formalized text corpus is then neither IR nor IE but a perfectly reasonable single request for an answer. "Tell me about fungi", addressed to a text corpus with an IR system, will produce a set of relevant documents but no particular answer. "Tell me what films my favorite movie critics likes", addressed to the right text corpus, is undoubtedly IE as we saw, and will produce an answer also. The needs and the resources available determine the techniques that are relevant, and those in turn determine what it is to answer a question as opposed to providing information in a broader sense.

At Sheffield we are working on two applications of IE systems funded as European Commission LRE projects: one, AVENTINUS is in the classic IE tradition, seeking information on individuals about security, drugs and crime, and using classic templates. the other ECRAN, a more research orientated project, searches movie and financial databases and exploits the notion we mentioned of tuning a lexicon so as to have the right contents, senses and so on to deal with new domains and relations unseen before.

In all this, and with the advent of speech research products and the multimedia associated with the Web, it is still important to keep in mind how much of our cultural, political and business patrimony is still bound up with texts, from manuals for machines, to entertainment news to newspapers themselves. The text world is vast and growing exponentially: one should never be seduced by multi-media fun into thinking that text and how to deal with it, how to extract its content, is going to go away.

References

P. M. Andersen, P. J. Hayes, A. K. Heuttner, L. M. Schmandt, and I. B. Nirenberg. Automatic extraction In

- Proceedings of the Conference of the Association for Artificial Intelligence, pages 1089-1093, Philadelphia, 1986.
- C. Aone, H. Blejer, S. Flank, D McKee, S Shinn. The Murasaki Project: Multilingual natural language understanding. In *Proceedings of the DARPA Spoken and Written Language Workshop*, 1993
- D. Appelt, J. Bear, J. Hobbs, D. Israel, and M. Tyson. (1992) SRI International FASTUS system MUC-4 evaluation results. *In Proceedings of the Fourth Message Understanding Conference (MUC-4)*, pages 143-147. Morgan Kaufmann, 1992.
- D. Appelt, J. Hobbs, J. Bear, D. Israel, M. Kanneyama, 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). Morgan Kaufmann.
- ARPA. The Tipster Extraction Corpus (available only to MUC participants at present). 1992.
- R. Basili, M. Pazienza, & P. Velardi, (1993) Acquisition of selectional patterns in sub -languages. Machine Translation, 8.
- Communications of the ACM: Special Issue on Text Filtering, 35(12), 1992.
- L. M. Carlson et al. The Tipster Extraction Corpus: A resource for evaluating natural language processing systems. / (In preparation.)
- K. Church, S. Young and G Bloothcroft (ed.) (1996) Corpus-Based Methods in Language and Speech, Dordrecht, Kluwer Academic Publishers.
- F. Ciravegna, P. Campia and A. Colognese. (1992) Knowledge extraction from texts by SINTESI, In *Proceedings of the 14th International Conference on Computational Linguistics (COLING92)*, pages 1244-1248, Nantes, France.
- J. Cowie, T. Wakao, L. Guthrie, W. Jin, J. Pustejovsky and S. Waterman. (1993) The Diderot information extraction system, In *Proceedings of the First Conference of the Pacific Association for Computational Linguistics (PACLING)*, Vancouver.
- J. Cowie, & W. Lehnert (1996) Information Extraction, in (Y. Wilks, ed.) Special NLP Issue of the Comm. ACM.
- H. Cunningham, R. Gaizauskas & Y. Wilks, (1995) GATE: a general architecture for text extraction, University of Sheffield, Computer Science Dept. Technical memorandum.
- DARPA. Proceedings of the Third Message Understanding Conference (MUC-3), San Diego, California, 1991.

 Morgan Kaufmann.
- DARPA. Proceedings of the Fourth Message Understanding Conference (MUC-4), McLean, Virginia, 1992. Morgan Kaufmann.
- B, Dorr, & D. Jones (1996) The role of word-sense disambiguation in lexical acquisition: predicting semantics from syntactic cues, Proc. COLING96.
- Gaizauskas, T. Wakao, K. Humphreys. H. Cunningham, and Y. Wilks (1995) Description of the LaSIE System as Used for MUC-6. *In Proceedings of the Sixth Message Understanding Conference* (MUC-6). DARPA.
- R. R. Granger (1977) FOULUP: a program that figures out meanings of words from context. Proc. Fifth Joint International Conference on AI.
- R. Grishman et al., (1996) Tipster Text Phase II Architecture Design, Proceedings of the Tipster Text Phase II

Workshop, Vienna, Virginia, DARPA

- C.F. Goldfarb (1990) The SGML Handbook, Clarendon Press, Oxford.
- J. Hobbs, D. Appelt, M. Tyson, J. Bear, and D. Israel. (1992) SRI International: Description of the FASTUS system. In Proceedings of the Fourth Message Understanding Conference (MUC-4), pages 268-275. Morgan Kaufmann.
- A. Kilgarriff (1993) Dictionary word-sense distinctions: an enquiry into their nature. Computers and the Humanities,
- G. F. DeJong. Prediction and substantiation: A new approach to natural language processing. *Cognitive Science*, 3:251-273, 1979.
- G. F. DeJong. An overview of the FRUMP system. In W. G. Lehnert and M. H. Ringle, editors, *Strategies for Natural P. S. Jacobs and L. F. Rau. SCISOR: Extracting information from on-line news. Communications of the ACM*, 33(11):88-97, 1990.
- W. Lehnert, C. Cardie, D. Fisher, J. McCarthy, E. Riloff, and S. Sonderland. University of Massachusetts:

 Description of the CIRCUS system. *In Proceedings of the Fourth Message Understanding Conference (MUC-4)*, pages 282-288. Morgan Kaufmann, 1992.
- W. Lehnert and B. Sundheim. A performance evaluation of text analysis technologies. *AI Magazine*, 12(3):81-94, 1991.

Language Processing, pages 149-176. Erlbaum, Hilldsale, N.J., 1982.

- B. Levin (1993) English verb classes and alternations, Chicago, IL.
- C. Mellish, A. Allport, R. Evans, L. J. Cahill, R. Gaizauskas, and J. Walker. The TIC message analyser. Technical Report CSRP 225, University of Sussex, 1992.
- R. Merchant, M. E. Okurowski and N Chichor (1996) The Multi-Lingual Entity Task (MET) Overview, *Proceedings of the Tipster Text Phase II Workshop*, Vienna, Virginia, DARPA
- W. Ogden and P. Bernick (1996) OLEADA: User-Centered TIPSTER Technology for Language Instruction, *Proceedings of the Tipster Text Phase II Workshop*, Vienna, Virginia, DARPA
- W. Paik, E.D. Liddy, E. Yu and M McKenna. Interpretation of proper nouns for information retrieval. In *Proceedings of the DARPA Spoken and Written Language Workshop*, 1993.
- P. Procter et al. (1994) The Cambridge Language Survey Semantic Tagger. Technical Report, Cambridge University Press
- P. Procter, editor. Longman Dictionary, of Contemporary English. Longman, Harlow, 1978.
- J. Pustejovsky and P. Anick (1988) On the semantic interpretation of nominals. Proc. COLING88.
- L. Rau. Extracting company names from text. In *Proceedings of the Seventh Conference on Artificial Intelligence Applications*, Miami Beach, Florida, 1991.
- E. Riloff and W. Lehnert. Automated Dictionary Construction for Information Extraction from Text, In *Proceedings* of the Ninth IEEE Conference on Artificial Intelligence for Applications, pages. 93-99. IEEE Computer Society Press. 1993.
- E. Riloff, and J. Shoen (1995) Automatically acquiring conceptual patterns without an annotated corpus, Proc. Third Workshop on Very Large Corpora.

- N. Sager. Natural Language Information Processing: A Computer Grammar of English and its Application. Addison-Wesley, Reading, Mass., 1981.
- R. C. Schank and R.P. Abelson. *Scripts, Plans, Goals and Understanding*. Lawrence Erlbaum Associates, Hillsdales, NJ, 1977.
- B. M. Sundheim and N. A. Chinchor. Survey of the Message Understanding Conferences. In *Proceedings of the DARPA Spoken and Written Language Workshop*, 1993.
- T. Wakao, R. Gaizauskas and Y. Wilks (1996), Evaluation of an algorithm for the recognition and classification of proper names. Proc. COLING96.
- Y. Wilks (1978) Making preferences more active, Artificial Intelligence, 11.
- Y. Wilks, B. Slator, B. & L. Guthrie (1996) Electric Words: dictionaries, computers and meanings. MIT Press.
- Y. Wilks (in press) Senses and Texts, Computational Linguistics.
- R. Weischedel. Studies in the statistical analysis of text, *Proceedings of the DARPA Spoken and Written Language Workshop*, pages 331. Morgan Kaufmann, 1991.
- R. Zajac and M. Vanni (1996) The Temple Translator's Workstation Project, *Proceedings of the Tipster Text Phase II Workshop*, Vienna, Virginia, DARPA