CS-97-10

Information Extraction: Beyond Document Retrieval

R. Gaizauskas and Y. Wilks

Information Extraction: Beyond Document Retrieval

Robert Gaizauskas and Yorick Wilks
Department of Computer Science
University of Sheffield
{robertg,yorick}@dcs.shef.ac.uk

Abstract

In this paper we give a synoptic view of the growth text processing technology of information extraction (IE) whose function is to extract information about a pre-specified set of entities, relations or events from natural language texts and to record this information in structured representations called templates. Here we describe the nature of the IE task, review the history of the area from its origins in AI work in the 1960's and 70's till the present, discuss the techniques being used to carry out the task, describe application areas where IE systems are or are about to be at work, and conclude with a discussion of the challenges facing the area. What emerges is a picture of an exciting new text processing technology with a host of new applications, both on its own and in conjunction with other technologies, such as information retrieval, machine translation and data mining.

1 Introduction: IE and IR

Information extraction (IE) is a term which has come to be applied to the activity of automatically extracting pre-specified sorts of information from short, natural language texts – typically, but by no means exclusively, newswire articles. For instance, one might scan business newswire texts for announcements of management succession events (retirements, appointments, promotions, etc.), extract the names of the participating companies and individuals, the post involved, the vacancy reason, and so on. Put another way, IE may be seen as the activity of populating a structured information source (or database) from an unstructured, or free text, information source. This structured database is then used for some other purpose: for searching or analysis using conventional database queries or data-mining techniques; for generating a summary; for constructing indices into the source texts.

Information extraction should not be confused with the more mature technology of information retrieval (IR), which given a user query selects a (hopefully) relevant subset of documents from a larger set. The user then browses the selected documents in order to fulfil his or her information need. Depending on the IR system, the user may be further assisted by the selected documents being relevance ranked or having search terms highlighted in the text to facilitate identifying passages of particular interest.

The contrast between the aims of IE and IR systems can be summed up as: IR retrieves relevant documents from collections, IE extracts relevant information from documents. The two techniques are therefore complementary, and their use in combination has the potential to create powerful new tools in text processing.

The differences and complementarity of the techniques can be illustrated by means of an example. The management succession event scenario outlined above was part of the DARPA MUC-6 information system evaluation (see section 2.2.4 below). For this evaluation texts pertaining to management succession were required. To obtain them, a corpus of Wall Street journal articles was searched using an IR system (*eg* [1]) with the query shown in Figure 1a). The query was deliberately *not* fine-tuned, as it was desired to obtain some proportion of irrelevant texts. A sample of a relevant text retrieved by this query is shown in Figure 1b). Such texts were then run through IE systems one of whose principal tasks was to fill in a template whose structure is shown in Figure 1c) to produce results as (partially) shown in 1d); as secondary output the system used here is able to generate a natural language summary of the information in the template as shown in e).

Not only do IE and IR differ in their aims, they differ in the techniques they employ. These differences arise partly from their difference in aim, but also for historical reasons. Most work in IE has emerged from research into rule-based systems in computational linguistics and natural language processing, while IR work, where it has not been *sui generis* has been influenced by information theory, probability theory, and statistics. Because of the requirement to extract information, IE must pay attention to the structural or syntagmatic properties of texts: 'Carnegie hired Mellon' is not the same as 'Mellon hired Carnegie' which differs again from 'Mellon was hired by Carnegie'. The simplest IR systems treat texts as no more than 'bags' of unordered words. More refined systems allow phrasal matching, proximity searching, and possibly thesaural expansion of query terms. But these techniques are still not adequate to extract, for example, role players in events and their attributes, as the following example shows:

1. 'BNC Holdings Inc. named Ms G. Torretta to succeed Mr. N. Andrews as its new chair-person';

2. 'Nicholas Andrews was succeeded by Gina Torretta as chair-person of BNC Holdings Inc.';

3. 'Ms Gina Torretta took the helm at BNC Holdings Inc. She succeeds Nick Andrews'.

To extract a canonicalised fact such as 'G. Torretta succeeds N. Andrews as chair-person of BNC Holdings Inc.' from each of these alternative formulations, some level of linguistic analysis is necessary – to cope with grammatical variation (active/passive), lexical variation ('named to' *vs.* 'took the helm'), and cross-sentence phenomena such as anaphora.

The inadequacies of IR techniques for getting at the content of texts, and hence their limitations in satisfying text users information needs, have been long known; indeed almost every paper on IE starts with a cry that IR is inadequate [2, 3, 4]. But is progress in IE being made? Are usable systems emerging, or is there a hope that they shortly will? Our aim in writing this paper is to give positive answers to these questions. In section 2 we review the history of IE, giving, if not an exhaustive review, at least a broad feeling for the work that has gone on in the area. In section 3 we try to give some flavour for the techniques and approaches that have been and are being used in IE systems, concentrating, excusably we trust, on the IE system we have developed and are currently using in a number of research projects. Then, in section 4 we discuss application areas and applied systems, where IE systems are actually performing real world tasks. We conclude, in section 5, by discussing some of the challenges facing IE in the future and the boundaries of IE. Overall we hope to give a reasonable picture of the achievements, limitations, and potential of this exciting new text processing technology.

2 A Brief History of Information Extraction

IE as an area of research interest in its own right was first surveyed in [4]. Very broadly one can say that the field grew very rapidly from the late 1980's when DARPA, the US

```
a)      chief executive officer head president chairman post succeed name

b)      <DOC>
        <DOCNO> 940413-0062. </DOCNO>
        <HL>    Who's News: @  Burns Fry Ltd. </HL>
        <DD> 04/13/94 </DD>
        <SO> WALL STREET JOURNAL (J), PAGE B10 </SO>
        <TXT>
        <p>
           BURNS FRY Ltd. (Toronto) -- Donald Wright, 46 years old, was named executive
        vice president and director of fixed income at this brokerage firm. Mr. Wright
        resigned as president of Merrill Lynch Canada Inc., a unit of Merrill Lynch & Co.,
        to succeed Mark Kassirer, 48, who left Burns Fry last month. A Merrill Lynch
        spokeswoman said it hasn't named a successor to Mr. Wright, who is expected
        to begin his new position by the end of the month.
        </p>
        </TXT>
        </DOC>

c)  <TEMPLATE> :=                  d)  <TEMPLATE-9404130062-1> :=
            DOC_NR:                        DOC_NR: "9404130062"
            CONTENT:                       CONTENT: <SUCCESSION_EVENT-9404130062-1>
    <SUCCESSION_EVENT> :=              <SUCCESSION_EVENT-9404130062-1> :=
            SUCCESSION_ORG:                SUCCESSION_ORG: <ORGANIZATION-9404130062-1>
            POST:                          POST: "executive vice president"
            IN_AND_OUT:                    IN_AND_OUT: <IN_AND_OUT-9404130062-1>
            VACANCY_REASON:                            <IN_AND_OUT-9404130062-2>
    <IN_AND_OUT> :=                        VACANCY_REASON: OTH_UNK
            IO_PERSON:                 <IN_AND_OUT-9404130062-1> :=
            NEW_STATUS:                    IO_PERSON: <PERSON-9404130062-2>
            ON_THE_JOB:                    NEW_STATUS: OUT
            OTHER_ORG:                     ON_THE_JOB: NO
            REL_OTHER_ORG:             <IN_AND_OUT-9404130062-2> :=
    <ORGANIZATION> :=                      IO_PERSON: <PERSON-9404130062-1>
            ORG_NAME:                      NEW_STATUS: IN
            ORG_ALIAS:                     ON_THE_JOB: NO
            ORG_DESCRIPTOR:                OTHER_ORG: <ORGANIZATION-9404130062-2>
            ORG_TYPE:                      REL_OTHER_ORG: OUTSIDE_ORG
            ORG_LOCALE:                <ORGANIZATION-9404130062-1> :=
            ORG_COUNTRY:                   ORG_NAME: "Burns Fry Ltd."
    <PERSON-9301190125-6> :=               ORG_ALIAS: "Burns Fry"
            PER_NAME:                      ORG_DESCRIPTOR: "this brokerage firm"
            PER_ALIAS:                     ORG_TYPE: COMPANY
            PER_TITLE:                     ORG_LOCALE: Toronto CITY
                                           ORG_COUNTRY: Canada
e)  BURNS FRY Ltd. named Donald Wright <ORGANIZATION-9404130062-2> :=
    as executive vice president.           ORG_NAME: "Merrill Lynch Canada Inc."
                                           ORG_ALIAS:  / "Merrill Lynch"
    Donald Wright resigned as president    ORG_DESCRIPTOR: "a unit of Merrill Lynch & Co."
    of Merrill Lynch Canada Inc..          ORG_TYPE: COMPANY
                                       <PERSON-9404130062-1> :=
    Mark Kassirer left as president of     PER_NAME: "Donald Wright"
    BURNS FRY Ltd.                         PER_ALIAS: "Wright"
                                           PER_TITLE: "Mr."
                                       <PERSON-9404130062-2> :=
                                           PER_NAME: "Mark Kassirer"
```

Figure 1: IR and IE: a) an IR query b) a retrieved text c) an empty template d) a fragment of the filled template e) a 'summary' generated from the filled template

defence agency, funded competing research groups to pursue IE. However, significant work of relevance was carried out before the DARPA initiative, some of it finding its roots in the 1960s. In this section we divide the work on IE into three broad categories: early work on template filling (work carried out or under way before the DARPA programme); work carried out in response to the DARPA MUC programme; and recent work on IE outside the DARPA programme. This division, like any for review purposes, is crude and not too much weight should be placed upon it.

2.1 Early Work on Template Filling

Applied work on filling structured records with information from natural language texts appears to have originated in two long-term, research-oriented natural language processing projects. The Linguistic String Project [5] at New York University began in the mid-60's and carried on into the 1980's. While concerned on the research side largely with the development of a large-scale computational grammar of English, the applications of the work were to do with deriving what Sager called information formats, regularised table-like forms which were, effectively, templates. These information formats abstracted away from the profusion of natural language forms and permitted a database to be defined against which 'fact retrieval' (as opposed to document retrieval) could be carried out. The applications were in the medical domain and concentrated on radiology reports and hospital discharge summaries. Some limited evaluation was carried out by contrasting the program's behaviour with the results of getting a human clinician to fill in a comparable information format solely on the basis of the information in the discharge summary. One interesting aspect of this work is that the information formats are *not* predefined *a priori* by experts in the field; rather, given a set of texts in a sub-language domain the information formats (the columns or fields in the tables) are induced by using distributional analysis to discover word classes in the domain (e.g. 'film shows clouding', 'x-rays indicate metastasis', etc. permit the definition of a `TEST | SHOW | MEDICAL FINDING` format). While inducing templates was abandoned through the 1980's and early 90's as simply too difficult, and the use of predefined, tailored templates created by domain experts adopted instead, there is renewed interest in automatically acquired templates [6].

The second long term project of relevance to the formation of IE as an autonomous area of research was the work on language understanding, and in particular on story comprehension, carried out at Yale University by Roger Schank and his colleagues [7, 8, 9, 10]. Central to this work was the notion that stories followed certain stereotypical patterns which Schank referred to as scripts. Knowing the script, language comprehenders are able to fill in details and make inferential leaps where the information required to make the leap is not present in the text. Thus a corporate merger, or a management succession event, or a doctor-patient examination all have predictable role-players and sub-events and knowing these permits us to make sense of a text describing any instance of such an event. The first attempt to build what might be called an IE system using this approach was made by one of Schank's students, Gerald De Jong, who designed and built a system called FRUMP [11]. It used what De Jong called sketchy scripts, a simplified version of the detailed scripts Schank had proposed, to process texts directly from a UPI news wire feed. De Jong's system employed sketchy scripts for sixty situations to extract information from news stories in domains ranging from earthquakes to labour strikes. The instantiated scripts were then used to generate summaries of the stories. His approach relied upon an alternation of predictor and substantiator modules which used, respectively, top-down, expectation-driven processing relying on predictions from the script and bottom-up, data-driven processing based on input from the text. This general approach has been adopted, in one way or another, by many IE systems since. De Jong's work is also notable for carrying out a reasonably extensive evaluation: six days of previously unseen news stories were fed in real-time through FRUMP and the results classified as to whether the stories were processed correctly, nearly correctly, wrongly, or were missed.

Following these initial projects, the 1980's saw the first commercial IE systems developed. The first system to be commercially deployed (to the best of our knowledge) was ATRANS, a system for automatic processing of money transfer messages between banks [12]. ATRANS adopted the Yale script-style approach to text processing, using script-driven predictions to identify actors (originating customer, originating bank, receiving bank, etc.) in order to fill in a template that was used, after human verification, to initiate automatic money transfers. Soon after, the Carnegie Group developed and deployed a 'fact extraction' system for Reuters called JASPER [13]. JASPER was designed to skim company press releases on PR Newswire and fill in a template containing information about company earnings and dividends. These templates were used to produce candidate news stories which were then validated or post-edited by journalists, offering them a significant savings in story preparation time. A final commercial system initiated in this period was the SCISOR system developed by GE for analysis of corporate mergers and acquisitions [2].

Two other academic research projects from this period should be mentioned. The first was a system developed by James Cowie to extract regularised descriptions (effectively, templates) of plants from wild flower guides [14]. Cowie's approach relied upon a domain-specific, handcrafted lexicon of keywords which allowed segments of the source text to be matched with appropriate sections of the target template. Rules pertaining to slots in the template (properties of plants) were then brought to bear on the selected portions of text and the property values extracted. The second was a project by G.P. Zarri to translate automatically French texts dealing with a particular period of French history into a 'metalanguage' which captured certain semantic relations pertaining to biographical details that were sought [15]. This metalanguage was organised around case frames for predicates, which can be viewed as small-scale templates: what was to be extracted were the roles in particular historical events, such as the naming to a position of an historical figure by a given body on a particular date at some location. The approach involved first using a syntactic analyser to establish the text's syntactic structure, and then carrying out semantic parsing in which lexical triggers – keywords in the domain – caused one or more of the case frames for key predicates to be invoked and then instantiated with material identified from the syntactic analysis, according to rules associated with the slots case frame slots.

2.2 The Message Understanding Conferences - MUC

2.2.1 Background to MUC

In the mid-1980's a number of sites in the US were working on IE from naval messages, in projects sponsored by the US Navy. In order to understand and compare their systems' behaviour better, a number of these message understanding (MU) projects decided to work on a set of common messages and then convene to see how their systems would perform when given some new, unseen messages. This gathering constituted the first of what has turned into an ongoing series of extremely productive message understanding conferences, or

MUCs, which have served as key events in driving the field of IE forward (the term 'message understanding' is now disappearing in favour of the more descriptively accurate 'information extraction')[16, 17, 18, 19].

There have been six Message Understanding Conferences to date and a seventh is planned for spring 1998. The objective of the conferences has been to establish a quantitative evaluation regime for IE or MU systems, which prior to these conferences had been sporadically assessed in an *ad hoc* fashion, frequently on the same data on which they had been trained. To date, the MUC conferences have been sponsored by DARPA and organised by the US Naval Command, Control, and Ocean Surveillance Center RDT&E Division (NRaD), formerly the Naval Ocean Systems Center, in San Diego, California.

A brief chronology and description of the MUCs is as follows:

MUC-1 Held in May 1987 in San Diego. Six systems participated. The texts were tactical naval operations reports on ship sightings and engagements. Twelve training reports were supplied, plus additional messages. Two unseen messages were distributed at the conference for participants to test their systems on. There was no task definition and there were no evaluation criteria.

MUC-2 Held in May 1989 in San Diego. Eight systems participated. Again the domain was tactical naval operations reports on ship sightings and engagements. 105 messages were supplied as training data and there were two test rounds, one with 20 blind messages and then, after system fixes, a second round of 5 blind messages just before the conference. This time a task was specified: a template was defined and fill rules for the slots supplied. Answer keys, i.e. correctly filled templates, were manually prepared for development and test texts. Resources in the form of lists of specialised naval terminology were also supplied. Evaluation criteria were defined, but by consensus deemed not to have been adequate. Scoring was done by participating sites.

MUC-3 Held in May 1991 in San Diego. Fifteen systems participated. The domain was newswire stories about terrorist attacks in nine Latin American countries. The stories were gathered from an electronic database but were originally items as diverse as newspaper stories, radio and television broadcasts, speeches, interviews, news conference transcripts, and communiqués. Most were translated from Spanish by the US Foreign Broadcast Information Service. 1,300 development texts were supplied and three blind test sets of 100 texts each were prepared. A template was defined consisting of 18 slots. Formal evaluation criteria were introduced, adapted from notions developed in information retrieval (specifically, precision and recall). A semi-automated scoring program was developed and made available for use by participants during development. Official scoring was done by the organisers.

MUC-4 Held in June 1992 in McLean, Virginia. Seventeen sites participated. The domain (Latin American terrorism) and template structures remained essentially unchanged. Changes were made to the task definition, corpus, measures of performance, and test protocols in order to provide greater focus on spurious data generation, to better assess system independence from training data, to make scoring more consistent, and to provide means for more valid score comparison between systems. This evaluation marked the beginning of the inclusion of the MUC conferences within the TIPSTER text programme [1]

MUC-5 Held in August 1993 in Baltimore, Maryland (coinciding with the TIPSTER-I 24-month evaluation). Seventeen systems participated (fourteen American, one British, one Canadian and one Japanese – this marked the first non-US involvement). Two domains – joint ventures in financial newswire stories and microelectronics products announcements – and two languages – English and Japanese – were tested. Substantial ancillary resources were supplied. Development and test corpora sizes were increased. Scoring was modified to include new evaluation metrics and the scoring program enhanced. More details of MUC-5 are presented in Section 2.2.3.

MUC-6 Held in November 1995 in Columbus, Maryland. Seventeen sites overall took part. The evaluation emphasized finer-grained evaluation and portability issues and comprised four subtasks – named entity recognition, coreference identification, and template element and scenario template extraction tasks. The domain of the scenario extraction task was management succession events in financial news stories. Sites were allowed to choose which subtasks they would undertake. MUC-6 is discussed further in section 2.2.4 below.

Across these evaluation exercises, the tasks have become progressively more difficult. Some effort was made to quantify this increase at MUC-5 and the conclusion drawn that there was an order-of-magnitude increase in task complexity on several measures between MUC-2 and MUC-5 [20]. Task complexity measures included text corpus complexity (e.g. vocabulary size, average sentence length), text corpus dimensions (e.g. volume of texts, total number of sentences/words), template characteristics (e.g. number of object types, number of slots), and difficulty of task (hard to measure, but considered, e.g., number of pages of relevance rules and template fill definitions). System performance has improved against this backdrop of increasing task complexity, indicating that genuine progress in developing this technology has been made in the past decade.

In sections 2.2.3 and 2.2.4 we describe MUC-5 and MUC-6 in some detail, as the most recent and most sophisticated IE evaluations.

2.2.2 Evaluation metrics

The evaluation metrics have evolved with each MUC. The starting points for the development of these metrics were the standard IR metrics of recall and precision. In the information extraction task, recall may be crudely interpreted as a measure of the fraction of the required information that has been correctly extracted and precision as a measure of the fraction of the extracted information that is correct. The definitions of these measures have been altered from those used in IR (but the names have been retained) to allow for overgeneration in IE where, unlike IR, data not present in the input can be erroneously produced.

Not only have recall and precision measures been redefined for the extraction task, but additional measures have been introduced as well. Slot fills can be correct, partially correct, or incorrect, but they can also be missing (no fill when there should be), spurious (fill present

[1] TIPSTER is a U.S. Government programme of research and development in the areas of IR and IE. TIPSTER is not an acronym and appears to have been adopted as a name because of the intelligence providing potential of these technologies (*cf.* the Oxford Concise Dictionary: **tipster** *n.* a person who gives tips, esp. about betting at horse-races.)

when it should not be), or non-committal (no fill when the answer key also contains no fill). These extra categories permit the introduction of measures of overgeneration (fraction of extracted information that is spurious), undergeneration (fraction of information to have been extracted that is missing), and substitution (fraction of the nonspurious extracted information that is not correct).

For MUC-3 and MUC-4 recall and precision were the primary metrics and the others were secondary. In addition, for MUC-4, van Rijsbergen's combined measure of recall and precision, the F-measure, was used [21]. But for MUC-5, recall and precision were deemed unofficial metrics and a new primary metric called error per response fill was introduced. This was an attempt to measure the fraction of a system's response that is 'wrong', i.e. the fraction of the combined actual and possible responses that were faulty. It was hoped that this measure would allow developers to focus more directly on the sources of their systems' difficulties, in particular on missing and spurious information which figures directly in the error-based metric, but only indirectly in the recall and precision metrics. In MUC-6 recall and precision regained their status as official metrics and the metrics were slightly modified so as to eliminate the category of partially correct slot fill. All of these metrics carried over to three of the four MUC-6 tasks, but only precision and recall metrics were employed for the coreference task and their definitions had to be modified to account for peculiarities of this task (see [22] for more details).

Since at least MUC-3, a text-filtering metric has also been employed to measure how good systems are at separating documents into relevant/nonrelevant categories. This measure operates at the level of texts as a whole (are templates generated for a given text when they should be or not) and not at the level of slots.

2.2.3 MUC-5

Task As with MUC-3 and MUC-4, the MUC-5/TIPSTER-I 24-month evaluation required systems to extract information from newswire stories. There were four possible tasks: two domains (joint ventures and microelectronics) and two languages (Japanese and English). These domain-language pairs are referred to using the acronyms EJV, JJV, EME and JME, in the obvious way. Participating non-TIPSTER-sponsored systems had to choose one domain and either or both languages; TIPSTER-sponsored systems were intended to operate in all four domain/language pairs. Most sites did only one task as this proved more than challenging enough. The EJV task was the most popular, and by common consent the most difficult; most of the following detailed remarks pertain to this task.

The MUC-5 template and fill rules were the most complex to date. For the first time the template was not a flat data structure, but rather allowed slots to contain pointers to other slots. Thus the template had an 'object-oriented' feel. For example, a joint venture was viewed as an object with various slots including its name and status ('existing', 'dissolved', etc), but also slots for the participating organisations, each of which was to be filled with a pointer to an organisation object, itself containing slots which in some cases contained pointers to other complex objects. In all there were 11 objects and 49 slots to be filled in. Slots were of four types: set fills (contained one of a given set of alternatives – e.g. organisation type could be company, person, government or other); string fills (contained a copy of some string from the original text – e.g. company name); normalised entries (contained data from the text transformed into a canonical form – e.g. dates, times, monetary amounts); references (pointers to other objects, as described above). As an indication of the level of detail required to define the extraction task, the fill rules occupied a 45 page document.

Resources There were three sources for the EJV materials: the Wall Street Journal, Lexus/Nexus, and PROMT. Roughly 2300 training texts were provided and answer keys were supplied for most of them. There was a dry run blind test set of 200 articles provided roughly half way through the evaluation, and a final blind test set of 286 articles. Official scoring was done for both dry run and final tests by MUC organisers but the scoring program was made available to all sites for use during development. This program was an extremely sophisticated piece of software which could be run in an entirely automatic mode, or in an interactive mode where the scorer is queried about the status of what the program judges may be partially correct answers.

The texts ranged in length from just two or three sentences, to several pages. Sentence lengths varied enormously, but some of length greater than seventy words were reported. In some places the texts contained tabular numeric data. The texts varied between mixed case and all upper case. All were originally marked up in SGML and contained certain reliably extractable information such as document id, date and source, flagged by SGML markers.

In addition to the training corpora and answer keys, considerable other data resources were supplied. These included: gazetteer of place names (246,908 entries); list of corporate names and nationalities (50,759 entries); list of corporate designators (133 entries); list of countries (244 entries); list of nationalities (216 entries); list of international organisations ($\sim$175 entries); definitions of (American) standard industry codes (17,779 entries); list of currency names/nationalities (217 entries); list of female forenames (4967 entries); list of male forenames (2924 entries); CIA world fact book. Some of the previous participants also made utility software available.

The methodology and effort required to produce the answer keys were both nontrivial. The production of the templates was undertaken by a small team of analysts, equipped with workstations and a software tool to aid in the extraction task. An elaborate procedure of selecting subsets of the documents to be multiply analysed was adopted in an attempt to ensure consistency in the answer keys. Of course the fill rules had to be modified as new complexity was uncovered and this required correcting previously created answer keys. The cost of producing the answer keys alone for MUC-5 and for the preceding TIPSTER extraction trials was more than $1 million US.

Results Table 1 shows the best raw score obtained in each of the four tasks discussed above. One interesting thing to note from these results is that in each domain the Japanese scores were higher. This observation has prompted discussion of whether in some sense Japanese is an easier language from which to extract information.

For error per response fill, undergeneration, overgeneration, and substitution the lower the score the better; for recall and precision the higher the score the better. Raw scores need to be interpreted very cautiously. Statistical studies were done on them [23] and for each task a number of ranks were identified within which raw score differences were claimed to be of no significance. For EJV there were 7 statistically significant ranks into which 13 systems were placed; in JJV 3 ranks for 5 systems; in EME 5 ranks for 7 systems; and in JME 2 ranks for 4 systems.

Task	ERR	UND	OVG	SUB	REC	PRE	P & R
EJV	61	30	39	19	57	49	52.8
JJV	50	32	23	12	60	68	63.8
EME	65	37	41	19	50	48	49.2
JME	58	30	38	14	60	53	56.3

Table 1: MUC-5 Best Overall Raw Scores indicating error per response fill (ERR), undergeneration (UND), overgeneration (OVG) substitution (SUB), recall (REC), precision (PRE) and combined precision and recall (P & R / F-measure) (from [18])

2.2.4 MUC-6

Tasks In MUC-6, rather than a single ‘end-to-end’ system evaluation as in MUC-5, participants were offered a menu of smaller evaluations from which they could pick and choose, depending on their interests and available resources. There were four evaluated tasks.

1. Named entity recognition. This task required the recognition and classification of definite named entities such as organisations, persons, locations, dates and monetary amounts. Classes of entity were reported by marking up the source text with SGML. In the usual MUC fashion, scoring involved comparing the system’s proposed result with manually prepared answer keys. Here is a simple example:

   ```
   <enamex type="organization">Bridgestone Sports Co.</enamex> said
   <timex type="date">Friday</timex> it has set up a joint venture in
   <enamex type="location">Taiwan</enamex> with a local concern and a
   Japanese trading house to produce golf clubs to be shipped to
   <pnamex>Japan</pnamex>.
   ```

   where `enamex` indicate an entity name, `timex` a time expression, and `pnamex` a place name expression.

2. Coreference resolution. This task required the identification of expressions in the text that referred to the same object, set or activity.

   Once again SGML markup was used to annotate coreferential expressions. For example

   ```
   <coref id="100">Galactic Enterprises</coref> said <coref id="101" type="ident"
   ref="100">it</coref> would build a new space station before the year 2016.
   ```

   The `id` attribute serves to identify arbitrarily, but uniquely, each string taking part in a coreference relation. The `ref` attribute indicates which string is coreferential with the one which it tags. The `type` attribute serves to indicate the relationship between anaphor and antecedent. The value `ident` for this attribute indicates identity, and in the final MUC-6 task definition was the only relationship to be marked. Other relationships such as `part-whole` and `set-member` had been considered, but were omitted due to difficulties in defining the task precisely enough.

   Coreference relations were only marked between certain syntactic classes of expressions (noun phrases and pronouns) and a relatively constrained class of relationships to mark was specified, with clarifications provided with respect to bound anaphors, apposition, predicate nominals, types and tokens, functions and function values, and metonymy.

3. Template element filling. This task required the filling of small scale templates wherever they occurred in the texts. There were only two such template elements, one for organisations and one for persons. These are illustrated in Figure 1.

4. Scenario template filling. The task required the detection of specific relations holding between template elements relevant to a particular information need (in this case corporate management personnel joining and leaving companies) and construction of an object-oriented structure recording the entities and details of the relation. This is illustrated in Figure 1.

The precise specifications of each of these tasks may be found in Appendices C-F of [19].

Four other evaluations had been considered, but were dropped due to lack of agreement over task definitions and lack of time and money for producing the development and test resources. These were parse structure evaluation (provide a canonical syntactic analysis of each sentence); predicate-argument structure evaluation (provide a canonical semantic analysis of each sentence); word sense disambiguation (disambiguate the sense of each open class, non-proper name word with respect to some standard lexical resource such as WordNet [24]); and cross-document coreference (determine coreferences between distinct documents).

The demand for this restructuring of the evaluation exercise arose for a number of reasons. Different participants had different interests and believed effort should be focussed in different areas. End-to-end systems IE were getting bigger and bigger and many research groups were excluded simply because they could not put the resources together to produce a massive system, where software engineering issues can soon come to eclipse research issues. Furthermore, comparison of systems and approaches had proved extremely difficult because the grain of the evaluation was too large. Finer scale evaluation, it was believed, would focus and promote more fruitful debate. However, it can be argued that any subdivision of the end-to-end IE task presupposes a processing approach to the task which may inhibit radically new approaches from emerging.

Resources As with MUC-5, the principal resources supplied by the organisers were annotated development and test corpora and scoring software. For both the dry run and final evaluations, 100 annotated development texts were provided for each of the four tasks. For the evaluations themselves there were 30 annotated test texts for the named entity and coreference tasks, and 100 annotated test texts for the scenario template and template element tasks. These texts were all Wall Street Journal texts, all of them mixed case. New scoring software was developed for the named entity and coreference tasks, and the MUC-5 scoring software enhanced for the template tasks.

Evaluation In MUC-6 the official evaluation metric reverted to precision and recall from the error-per-response-fill metric used in MUC-5. These two metrics had shown themselves to be very closely in line in MUC-5 and participants generally preferred precision and recall (perhaps because one tries to maximise these measures, whereas one tries to minimise error-per-response-fill, which casts the whole exercise in a more negative light).

The two template filling tasks were scored as in previous MUCs, with improvements to the scoring software, but no major departures. The named entity task required a new scorer based on comparing SGML-marked up strings, but the standard definitions of recall and precision carry over quite naturally here. However, in the coreference task, a problem arises

Task	ERR	UND	OVG	SUB	REC	PRE	P & R
Named Entity	5	2	1	2	96	97	96.42
Coreference (High Recall)					63	63	
Coreference (High Precision)					59	72	
Template Element	29	20	5	8	74	87	79.99
Scenario Template	57	41	12	20	47	70	56.40

Table 2: MUC-6 Best Overall Raw Scores indicating error per response fill (ERR), undergeneration (UND), overgeneration (OVG) substitution (SUB), recall (REC), precision (PRE) and combined precision and recall (P & R / F-measure) (from [19])

which requires that the precision and recall scoring measures be specially adapted. Clearly, more than two markables may corefer, i.e., there may be chains of coreferences, not simply coreferential pairs. In the case of chains, how to record the chain and how to score systems which fail to discover all the links in the chain become central issues. See [22] for a full discussion of the definitions of precision and recall for the coreference task.

Results Table 2 shows the best raw score obtained in each of the four tasks. In all but the coreference case the results of the system with the best combined precision and recall score (F-measure) have been displayed (thus, there may be other systems which obtained higher scores on one of the other measures). Due to differences in the approach to scoring the coreference task and the other tasks, only recall and precision measures were available for coreference, and no satisfactory combined measure could be defined.

2.2.5 An Assessment of MUC

Even after doing statistical significance studies it is hard to come to any firm conclusion about the superiority of a given approach, principally because of the varying levels of resources that different sites brought to the task – person-months spent on development, qualifications and backgrounds of the people doing the development, software and hardware resources committed, and so on. At the conference every site could put up a graph showing a steep line of improvement from the immediately preceding dry run evaluation and claim (especially to their funding bodies !) that given another few months they could make spectacular gains. Clearly this improvement has to stop somewhere; but there is no way of telling which approach will level out when and at what level.

Another criticism frequently made of the MUC evaluations is that they lead to copy-cat behaviour, whereby systems tend to converge upon the same approach because any advantage is quickly picked up by others afraid to lag behind in the short term because of funding implications of being seen to be a ‘loser’.

Each of these criticisms can be at least partially answered. The first one – that the evaluation results do not let us draw unequivocal conclusions – by observing that imperfect evaluation is better than none at all. The results can tell us important things; we simply need to be careful in interpreting the results. The second criticism – that participating sites tend to play safe by copying successful approaches – may be true of some sites (perhaps those directly dependent on linked funding), but is certainly not true of all sites, particularly academic ones (section 3.3.1 gives some indication of the wide range of approaches still being entertained). Besides the rapid transfer of successful technology can hardly be viewed as completely deleterious.

In all the MUC evaluations have provided the IE community resources, evaluation tools, and perhaps above all a sense of identity and a forum for exchange of ideas. There may come a time when their utility becomes questionable; but they have proved of significant worth to date.

2.3 Other Work on Information Extraction

The MUC evaluations are still running, but concurrent with them, either unrelatedly or in part because of the higher interest in IE they have generated, numerous other IE projects can be identified. This list describes some significant European IE projects, but is almost certainly incomplete given the rapidly expanding nature of the field.

Two projects which started in the late 1980's illustrate the use IE systems for processing sublanguages – specialised languages that are developed within a restricted area of human activity and which are frequently characterised by extragrammaticality (from the perspective of the 'mother' language), idiosyncratic lexical forms, and heavy use of ellipsis (because of the shared world knowledge which the context which gives rise to the sublanguage supplies). The first of these is the POETIC (Portable Extendable Traffic Information Collator) system [25] whose function was to extract information about road traffic incidents causing traffic congestion from police incident logs and to generate advisory bulletins to be broadcast to motorists. Police incident logs form a sublanguage in the sense defined above, and the system utilised a special grammar and lexicon, as well as a domain-specific reasoning component to deal with the highly telegraphic and idiosyncratic forms found in the police logs.

The second system was SINTESI (Sistems INtegrato per TESti in Italiano) which processed short texts describing car faults and filled in a template identifying the main fault, chain of causes, chain of effects, car parts involved etc. [3]. Once again, because of the nature of the sublanguage, the approach relied extensively on domain-specific lexical-semantic knowledge (caseframes for relevant objects in the domain).

The Language Engineering (LE) initiatives within the Commission of the European Communities (CEC) Third and Fourth Framework programmes have supported a number of IE projects, several of which are currently underway. These are simply listed with references for the interested reader, as there is not space to describe them, and in some cases, as the projects are just underway, there is yet little published material about them. The TREE (TRans European Employment) project aims to make information available to job seekers across the European Union by extracting job details from electronic job advertisements and storing them in a database which can be browsed by job seekers in their own language [26, 27]. The FACILE (Fast Accurate Categorisation of Information using Language Engineering) project, following on from the COBALT project aims to categorise and filter news stories of interest to stock market traders, using extraction-like techniques [28, 29, 30]. Finally, at Sheffield we are working on two applications of IE systems within the CEC LE projects: one, AVENTINUS is in the classic IE tradition, seeking information on individuals about security, drugs and crime, and using classic templates [31, 32]. 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 [33].

3 Approaches to Information Extraction

Since IE systems are large, complex software systems usually consisting of many components, classifying them is not an easy task. Perhaps the most useful aid in this task is a description of the generic IE system provided by J. Hobbs [34]. His description allows newcomers to the field to grasp the principal processing stages involved in IE and provides IE system developers with a standard system description against which to differentiate their own. While this description was derived as a synthesis of the approaches used in MUC-4 systems, it remains broadly true.

Armed with this general description we then turn to a description of the LaSIE (Large Scale Information Extraction) system which we have developed at Sheffield, using the system we know best to illustrate in more detail the sorts of processing involved in information extraction. While LaSIE is quite distinct from many IE systems, it is not difficult to see how it fits Hobbs's general rubric. Following this moderately detailed description of how one IE system works, we conclude this section with a discussion of some of the general trends that are currently influencing the direction of IE system development.

3.1 The Generic IE System

Hobbs describes the generic IE system as a "cascade of transducers or modules that at each step add structure and often lose information, hopefully irrelevant, by applying rules that are acquired manually and/or automatically" ([34], p. 87). To describe such a system requires identifying the modules, identifying each module's input and output, identifying the form of the rules the modules apply, and specifying how the rules are applied and how they are acquired.

According to Hobbs, a typical IE system consists of a sequence of ten modules:

1. Text Zoner. Divides the input text into a set of segments.

2. Preprocessor. Converts a text segment into a sequence of sentences, where each sentence is a sequence of lexical items, with associated lexical attributes (e.g. part-of-speech).

3. Filter. Eliminates some of the sentences from the previous stage by filtering out irrelevant ones.

4. Preparser. Detects reliable small-scale structures in sequences of lexical items (e.g. noun groups, verb groups, appositions).

5. Parser. Analyses a sequence of lexical items and small-scale structures and attempts to produce a set of parse tree fragments, possibly complete, which describes the structure of the sentence.

6. Fragment Combiner. Turns a set of parse tree or logical form fragments into a parse tree or logical form for the whole sentence.

7. Semantic Interpreter. Generates a semantic structure or meaning representation or logical form from a parse tree or parse tree fragments.

8. Lexical Disambiguation. Disambiguates any ambiguous predicates in the logical form.

9. Coreference resolution or discourse processing. Builds a connected representation of the text by linking different descriptions of the same entity in different parts of the text.

10. Template generator. Generates final templates from the semantic representation of the text.

Of course not all systems exhibit all of these modules, nor do they necessarily perform their processing in exactly this sequence (in particular stages 6 and 7 may occur in the reverse order).

3.2 LaSIE: A Case Study

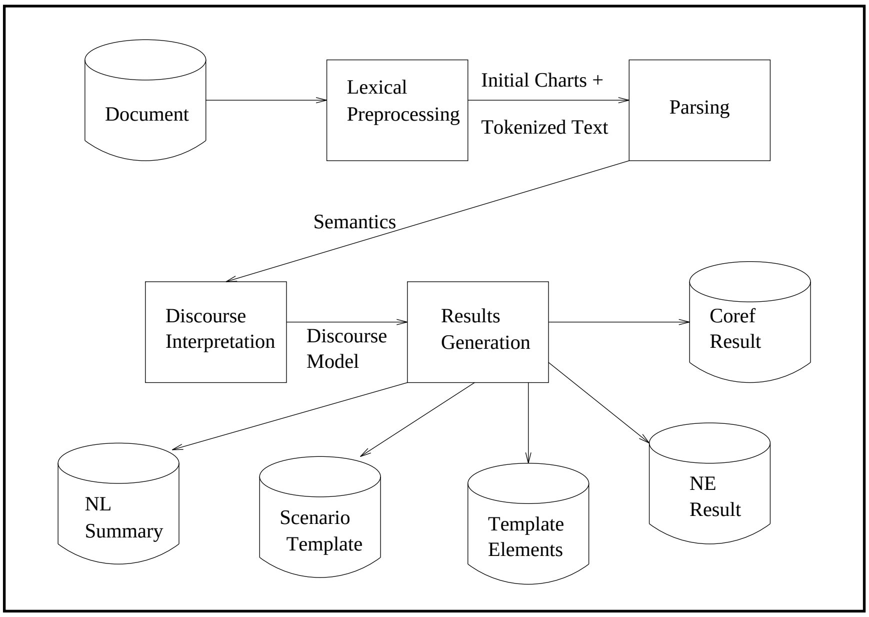

Figure 2: LaSIE System Architecture

LaSIE was designed as a general purpose IE research system, initially geared towards, but not solely restricted to, carrying out the tasks specified in MUC-6: named entity recognition, coreference resolution, template element filling, and scenario template filling. In addition, the system can generate a brief natural language summary of any scenario it has detected in the text. All of these tasks are carried out by building a single rich model of the text – the discourse model – from which the various results are read off.

The high level structure of LaSIE is illustrated in Figure 2. The system is a pipelined architecture which processes a text one sentence at a time and consists of three principal processing stages: lexical preprocessing, parsing plus semantic interpretation, and discourse interpretation. The overall contributions of these stages may be briefly described as follows:

lexical preprocessing reads and tokenises the raw input text, tags the tokens with parts-of-speech, performs morphological analysis, performs phrasal matching against lists of proper names;

parsing and semantic interpretation builds lexical and phrasal chart edges in a feature-based formalism then does two pass chart parsing, pass one with a special named entity grammar, pass two with a general grammar, and, after selecting a 'best parse', constructs a predicate-argument representation of the current sentence;

discourse interpretation adds the information from the predicate-argument representation to a hierarchically structured semantic net which encodes the system's world model, adds additional information presupposed by the input, performs coreference resolution between new and existing instances in the world model, and adds any information consequent upon the new input.

Subsequent to MUC-6, LaSIE was re-engineering at the architectural level to make it function within a language engineering research architecture called GATE – the General Architecture for Text Engineering also developed at Sheffield. GATE is a software environment that supports researchers who are working in natural language processing and computational linguistics and developers who are producing and delivering language engineering systems [35, 36]. It is based on the TIPSTER architecture [37], an object-oriented data model designed to support a broad range of document processing tasks and promoted as a standard for the information retrieval and extraction tasks within the DARPA-sponsored TIPSTER text programme. The re-engineered LaSIE system functioning within GATE is called VIE (Vanilla IE system). It was derived from LaSIE by standardising LaSIE module interfaces so that all modules communicated with each other via the GATE document manager (allowing for easy substitution of improved modules with similar functionality – e.g., better part-of-speech taggers, or parsers). Further details of LaSIE and VIE can be found in [35, 38]. [2]

The processing of the system is best illustrated by means of an example. We will discuss what processing goes on in each of the three principal stages identified above with respect to the small text shown in Figure 1b).

3.2.1 LaSIE: Lexical Processing

This stage comprises five modules.

1. Tokenisation. This module does both text segmentation and tokenisation. In the example text it distinguishes the document header (everything preceding the `<TXT>` tag) from the document body, and in longer texts would segment the text into paragraphs. Tokenisation involves identifying which sequences of characters will be treated as individual tokens – for example, treating SGML tags as single tokens, but separating other punctuation from preceding characters (so `<TXT>` is a token but `Ltd.,` in the first line of the text is three tokens).

2. Sentence splitting. This module determines sentence boundaries in the text – a non-trivial task as full stops are not sufficient guides. For example, they may occur in names (`Allan J. Smith`) and after abbreviations (`Inc. Mr.`), though of course the latter may end sentences too.

3. Part-of-speech tagging. We have used a modified version of the rule based part-of-speech tagger developed by E. Brill [39]. It processes one sentence (sequence of tokens) at a time and associates with each token one of the forty-eight part-of-speech tags in the University of Pennsylvania tagset [40]. Thus, for input such as `Donald Wright, 46 years old` the tagger produces output of the form `Donald/NNP Wright/NNP ,/COMMA 46/CD years/NNS old/JJ`, where `NNP` designates a proper noun, `CD` a cardinal number, `NNS` a plural common noun, and `JJ` an adjective.

[2] GATE and VIE are both publicly available: see `http://www.dcs.shef.ac.uk/research/groups/nlp/gate` for details.

4. Morphological analysis. This module does a limited form of morphological analysis, determining root forms of nouns and verbs. In our example `years` will analysed as having root `year` and affix `s` and `named` would be analysed as having root `name` and affix `ed`.

5. Gazetteer lookup. We employ 5 gazetteers, or lists of names, to facilitate the process of recognising and classifying named entities. These are organisation names, location names, personal given names, company designators (`Corp.`, `Ltd.`, etc.), and personal titles (`Mr.`, `President`), etc. In our example text, `Toronto` and `Canada` are tagged as places, `Donald` and `Mark` as first names, `executive vice president` and `president` as personal titles and `Ltd.`, `Inc.` and `Co.` as company designators. Only well known names are stored in these lists, so, for example, while `Merrill Lynch` and `Burns Fry` are prestored, a company such as `Sheffield Motor Repairs` would not be.

   In addition we use four lists of trigger words, to tag words which occur inside multi-word proper names, and which reliably permit the class of the proper name to be determined. For example, 'Wing and Prayer Airlines' is almost certainly a company, given the presence of the word Airlines; 'Bay of Pigs' almost certainly a location given the word Bay. This and further aspects of the system's algorithm for proper name recognition are discussed further in [41].

3.2.2 LaSIE: Parsing

The parsing and semantic interpretation stage of LaSIE is carried out by a single module. However this stage consists of three substages. The first substage is parsing with a special named entity grammar. We use a bottom-up chart parser [42] and a manually constructed context-free grammar of 177 rules pertaining to named entities to recognise multi-word structures which identify organisations, persons, locations, dates, and monetary amounts. For example, a rule like `ORGAN_NP --> ORGAN_NP LOC_NP CDG` allows us to recognise the organisation name `Merrill Lynch Canada Inc.` and a rule like `PERSON_NP --> FIRST_NAME NNP` allows us to recognise the person name `Donald Wright`. Semantic interpretation is carried out in parallel with parsing. This amounts to assigning a regularised form in a predicate-argument notation to each phrase identified by the grammar. For proper names this logical form consists of two terms, a unary predicate specifying the type of the entity and a binary predicate specifying the actual name string. For example, `Burns Fry Ltd.`, following its syntactic analysis, is assigned the logical form `organization(e17), name(e17,'Burns Fry Ltd.')` where `e17` is a unique new identifier introduced to provide an unambiguous handle for the entity referred to in the text as `Burns Fry Ltd.`.

The second substage is parsing with a more general phrasal grammar. The same parser mechanism is used, but this time with a grammar of 110 rules designed to recognise noun phrases, verb phrases, prepositional phrases, adjectival phrases, sentences, and relative clauses. This grammar was extracted from a large manually annotated corpus of newswire text, the Penn Treebank [40], using a set of programs designed for the purpose [43]. Again, a semantic interpretation is built up during parsing. For instance the sentence `Donald Wright, 46 years old, was named executive vice president and director of fixed income at this brokerage firm` is parsed and assigned a top level structure as shown in figure 3. Note that this analysis is partial due to lack of coverage in the grammar; however, this does not prevent useful information from being derived. From the structural relations that are identified a logical form may assigned. For key parts of this sentence this takes the form:

```
person(e21), name(e21, 'Donald Wright')
name(e22), lobj2(e22,e23)
title(e23,'executive vice president')
firm(e24), det(e24,this)
```

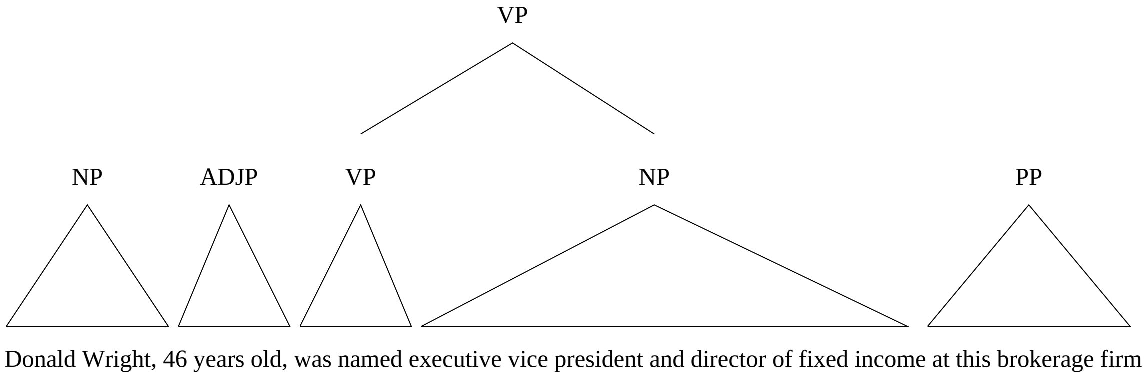
VP
NP
ADJP
VP
NP
PP
Donald Wright, 46 years old, was named executive vice president and director of fixed income at this brokerage firm

Figure 3: A LaSIE Parse Forest

Despite the fact that the parser is complete, *i.e.* finds all structural analyses of its input sentence according to the grammar, it is rare that these analyses contain a unique, spanning parse of the sentence. Consequently, the final substage of the parsing module involves selecting a "best parse" from the set of partial, fragmentary, and possibly overlapping (and hence incompatible) phrasal analyses which the parser has found. This is currently done by choosing that sequence of non-overlapping phrases of semantically interpretable categories (sentence, noun phrase, verb phrase and prepositional phrase) which covers the most words and consists of the fewest (hence largest) phrases.

3.2.3 LaSIE: Discourse Processing

The principal task of the discourse processing module in LaSIE is to integrate the semantic representations of multiple sentences into a single model of the text from which the information required for filling a template may be derived. The discourse processor works on the semantic representations passed onto it from the parser, though these include a record of the surface text from which they were derived, and in particular permit the order in which entities were introduced to be recovered.

The discourse interpretation stage of LaSIE relies on an underlying 'world model', a declarative knowledge base that both contains general conceptual knowledge and serves as a frame upon which a discourse model for a multi-sentence text is built. This world model is expressed in the XI knowledge representation language [44] which allows straightforward definition of cross-classification hierarchies, the association of arbitrary attributes with classes or individuals, and the inheritance of these attributes by individuals.

The world model consists of an ontology plus an associated attribute knowledge base. In LaSIE the ontology consists mostly of classes or 'concepts' directly relevant to a specific template filling task. So, for example, for the management succession scenario the ontology is constructed to contain details about persons, posts, and organisations, and also about events involving persons leaving or taking up posts in organisations.

Associated with each node in the ontology is an attribute-value structure. Attributes are simple `attribute:value` pairs where the value may either be fixed, as in the attribute `animate:yes` which is associated with the `person` node, or where the value may be dependent on various conditions, the evaluation of which makes reference to other information in the model. Certain special attribute types, `presupposition` and `consequence`, may return values which are used at particular points to modify the current state of the model, as described in the following section. The set of attribute-value structures associated with the whole ontology is referred to as the attribute knowledge base.

The higher levels of the ontology for the MUC-6 management succession extraction task are illustrated in figure 4, along with some very simple attribute-value structures.

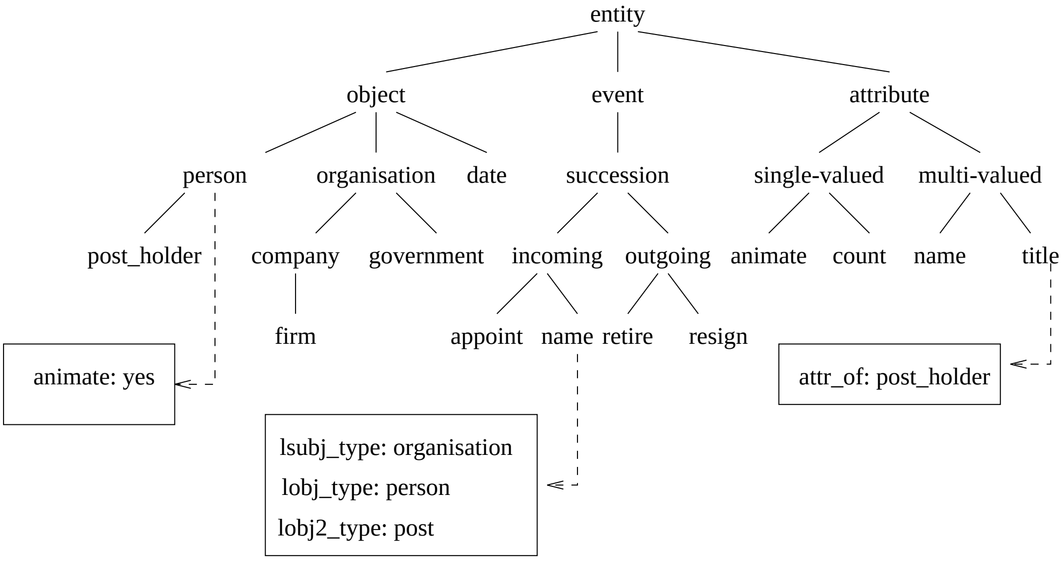

Figure 4: A Fragment of the LaSIE World Model and Associated Attribute Knowledge Base

The world model described above can be regarded as an empty shell or frame to which the semantic representation of a particular text is added, populating it with the instances mentioned in the text. The world model which results is then a model specialised for the world as described by the current text; we refer to this specialised model as the discourse model.

Figure 5 illustrates how instances are added to the world model, specialising it to convey the information supplied in a specific text. In the figure instances are indicated with the notation `e20, 21`, etc. and are shown connected by dashed lines to their classes. The figure reflects the state of discourse processing part way through the interpretation of the sentence 'Donald Wright, 46 years old, was named executive vice president and director of fixed income at this brokerage firm', as will be described below. Instances shown in bold derive from previous text (just `e20` in this case, derived from the dateline), instances in normal font indicate entities deriving directly from the current sentence, and those in italic font (just *e25* here) are instances hypothesised in processing the current sentence.

Discourse processing proceeds in four substages for each new sentence representation

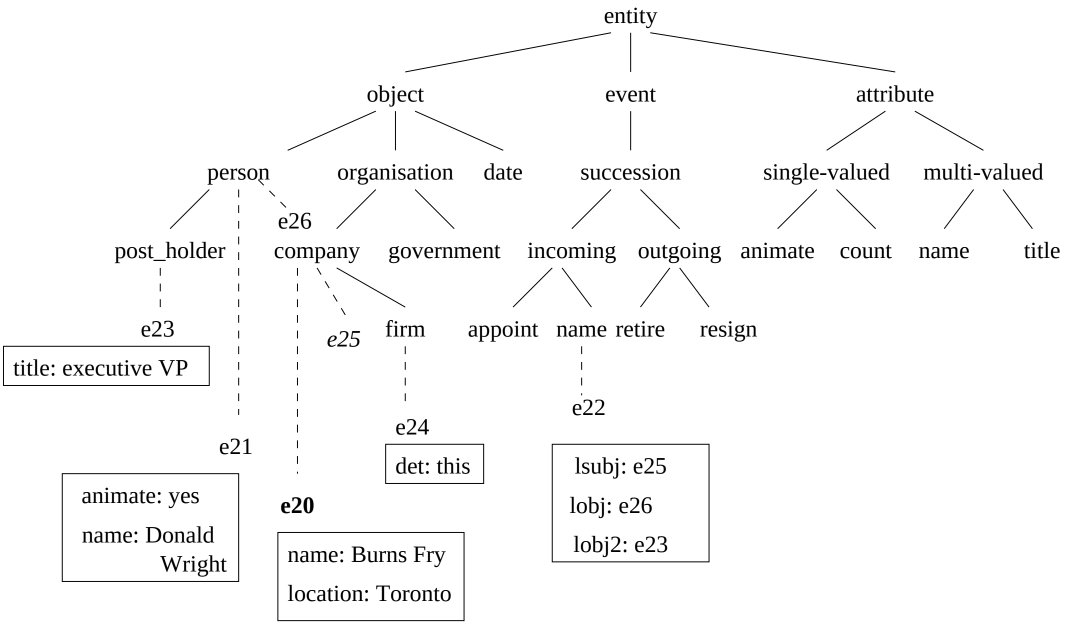

Figure 5: A Fragment of the LaSIE Discourse Model

passed on from the parser. First, the semantic representation produced by the parser is processed by adding its instances, together with their attributes, to the discourse model which has been constructed so far for the text. Instances which have their semantic class specified in the input (via unary predicates) are added directly to the discourse model, beneath their class in the ontological hierarchy (e.g. `firm(e24)`). Attributes – binary predicates in which the first argument is always an instance identifier – are added to the attribute-value structure associated with instance identifiers occurring within them, provided the class of the instance is known.

In the second stage, presuppositions are expanded, leading to further information being added to or removed from the model. In the current example, this has two effects. First, it permits missing semantic class information for instances to be derived from type restrictions on attribute arguments. For instance, an `attr_of` attribute associated with the node in the ontology corresponding to the `title` attribute, records that this attribute holds only of entities of type `post_holder`. Thus, given the input fact `title(e23,executive VP)` but no input fact specifying the class of `e23`, it becomes possible to attach the instance `e23` beneath the correct class in the ontology. Second, the semantic types of verbal roles are used to hypothesise entities which fulfil those roles, if they are not present, or have not been discovered, in the input. In this case the fact that 'Donald Wright' is the logical object of the 'was named' event has not been determined by the parser, as the intervening phrase '46 years old' was not properly parsed, hence preventing the parser from identifying 'Donald Wright' as the surface subject/logical object of the passive verb phrase. Thus, a person `e26` is added to the model to play this role. In a similar fashion `e25`, an organisation, is added to the model to play the role of the logical subject of the naming event.

The third stage involves comparing all new instances (those introduced by this sentence) with previously existing instances to determine whether any pair can be merged into a single instance, representing a coreference in the text. The algorithm takes into account considerations such as the instances' textual proximity and the consistency of their semantic classes and attributes. For the current example the coreference algorithm leads to the merging of `e26` and `e21` – that is, 'Donald Wright' is recognised as the logical object of the naming event – and `e25` is merged with `e24` – that is, 'this brokerage firm' is identified as the logical subject of the naming event. Subsequently these merged entities are merged with `e20` – that the brokerage firm doing the naming is identified as 'Burns Fry'. The reader is referred to [45] for further details, and an evaluation, of the coreference algorithm.

The final stage of discourse processing is consequence expansion. This stage is intended to allow any inferences to be drawn which can now be made given the addition to the discourse model of the information in the current sentence. Its primary use in LaSIE is to allow inference rules associated with template objects and slots to infer values for these objects and slots from information now present in the discourse model.

After all sentences in a text have been processed, the template will have been filled to the best of the system's abilities. The template is then written out in whatever form is required.

3.3 Trends

IE is not an isolated activity and is being influenced by and is in turn influencing other activities in natural language processing and computational linguistics. In this section we look briefly at three trends that can be seen in the recent development of IE: the movement towards shallower processing (or towards what might be called an 'appropriate' level of processing for the task), the movement away from handcrafted rule sets towards automatically acquired rule sets, and the movement towards coupling together relatively independent modules. Of course these trends are not entirely independent. They are all part of a general move towards a more empirically oriented approach to NLP that has emerged for a host of reasons, including the availability of large scale electronic corpora, frustration with theoretical developments that seemed to be losing touch with the reality of the data, and the drive towards applications.

3.3.1 Shallow vs Deep Processing

Given the pragmatic constraints imposed by the IE task – the relatively limited understanding required – many developers of IE systems have, in recent years, opted for engineering solutions that de-emphasize the substantial body of theoretical work both in computational syntax and semantics and in knowledge representation and reasoning. This de-emphasis is perhaps most dramatically illustrated by SRI who abandoned, quite consciously, the theoretically motivated TACITUS system after MUC-3 (1991) in favour of the pragmatically motivated FASTUS system which they have used for MUC-4 (1992) through MUC-6 (1995). TACITUS [46] attempted a full syntactic analysis, using a large scale grammar of English, performed semantic interpretation to produce first-order predicate calculus representations, and then used abductive reasoning to interpret the semantic representations of individual sentences in the context of a schema pertaining to the scenario of interest. FASTUS [47, 48, 49], by contrast, uses a cascade of finite-state transducers that successively tokenise, recognise names, recognise phrases, recognise template patterns, and then combine or merge partially filled templates to generate the final template. SRI have been keen to stress that this change in direction has not happened because they concluded that the TACITUS approach was faulty, but because they believed it was inappropriate for the task. TACITUS did text understanding, FASTUS information extraction, the latter, on their view, a much simpler task that does not require the theoretical and computational sophistication of TACITUS. The chief gain from the switch has been speed (from 36 hours to 12 minutes for 100 texts between MUC-3 and MUC-4) and to some extent ease of porting to new domains. Though performance results, in terms of combined precision and recall, are not strictly comparable between MUCs, it is worth noting that FASTUS scores surpassed TACITUS scores by about 16% between MUC-3 and MUC-4, mostly due to increased recall.

SRI have not been alone in moving away from a more powerful, linguistically motivated approach towards a more restricted, task-specific, engineering-driven approach. Recent IE systems developed by General Electric, Mitre Corporation, New York University and SRA have all come to be considered exemplars of a 'shallow' processing approach to IE which promises, if not better recall and precision, at least faster, more portable systems.

This movement away from the more theoretically motivated work of the 1980's has engendered considerable debate (and rhetoric) about 'shallow' versus 'deep' approaches to information extraction. This debate is ongoing and the underlying distinction, while reflecting important insights, needs to be analysed, as it can lead to distortion and over-simplification. In particular, it is important to distinguish at least two ways in which processing in an IE system can be shallower or deeper. The processing in an IE system can be divided coarsely into two parts: the syntactic portion that works on single sentences of the input and the discourse-level portion that integrates information from the syntactic analyses of multiple sentences. The former typically includes tokenisation, part-of-speech tagging, phrasal pattern matching or parsing and produces a regularised form which may be anything from a partially filled template to a full logical form. The latter takes whatever regularised form has been produced by the former and, perhaps using more general knowledge of domain, attempts to integrate information from the individual sentence representations into a larger scale structure which ultimately either is, or serves to provide, the information for the final template.

Thus, processing in an IE system can be shallower or deeper depending on the shallowness or depth of processing in each of these two processing stages. First, the syntactic analysis the system performs can be more or less thorough. At one extreme there are systems which employ formally weak mechanisms (finite-state pattern matchers) to apply domain-specific lexically-triggered patterns; at the other extreme there are systems which employ formally stronger mechanisms (complete parsers for context-free or even more expressive formalisms) to apply general grammars of natural language. Examples of the former include the SRI FASTUS system, Mitre's Alembic system [50], and the SRA [51] and NYU [52] MUC-6 systems; examples of the latter include the TACITUS system mentioned above, the Proteus system [53], and the PIE system [54]. Systems like LaSIE and the BBN PLUM system [55] which use a domain independent grammar, but only attempt fragmentary parsing, fall somewhere in the middle.

Second, the discourse or multi-sentence level processing can be more or less general. Thus, the semantic representation derived from the syntactic analysis can be expressed in a more or less general formalism and manipulated by more or less general algorithms which attempt to integrate it into a more or less general model of the text and domain. There may or may not be any attempt to use declaratively represented world and domain knowledge to help in resolving ambiguities of attachment, word sense, quantifier scope, and coreference, or to support inference-driven template filling. At one extreme there are information extraction systems which produce semantic representations that are fragments of the target template for just those sentences that yield template relevant information and then merge these using *ad hoc* heuristics to produce the final template (e.g. FASTUS and the SRA MUC-6 system); at the other extreme there are systems that use abductive theorem provers and axiomatisations of the domain to compute the least cost explanation of the first order logic expressions derived from every sentence in the input and then generate the template from the resulting underlying logical model (e.g. TACITUS). In between lie systems that translate their input into some sort of template-independent predicate-argument notation and use some amount of declaratively represented information about the domain to assist in doing coreference and inference driven template filling. LaSIE falls into this camp as do the NYU MUC-6 system and the MITRE Alembic system.

3.3.2 Hand-crafted Rules vs Automated Rule Acquisition

Early successful systems like JASPER (see section 2.1 above), depended on very complex hand-crafted templates, made up by analysts. 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 generalisation from the large volumes of text data that machines can now manipulate.

A conspicuous success has been part-of-speech taggers, systems that assign one and only one part-of-speech symbol to a word in a running text and do so on the basis (usually) of statistical generalisations across very large bodies of text. Recent research 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, and attaching word sense tags to words in texts to disambiguate them in context.

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 [56].

What such lexicons lack is a dynamic view of a language; they are inevitably fossilised 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 analysed 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 them to adapt to new domains and corpora, which will mean dealing with obsolescence and with the specialised vocabulary of a domain not encountered before.

3.3.3 Modularisation

As noted above there has been a movement away from theory prescribed modules whose processing is controlled by sets of handcrafted rules towards data-dependent modules whose processing is controlled by rules or parameters acquiring from automatically analysing large text corpora. These modules include part-of-speech tagging, text-alignment in different languages, syntax analysis, word sense disambiguation and so on. Aside from the fact that their rules or parameters are acquired automatically, the other striking thing about these modules is their independence: 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 – against their specifications. Recently there has been a move to support this kind of modularisation explicitly through the development of text processing architectures like the TIPSTER architecture [57] and implementations of it like the General Architecture for Text Engineering (GATE) [35, 36]. These architectures support rapid addition and interchange of modules and represent a commitment to a modular approach to language engineering.

While language engineering modules can be developed and evaluated independently it is important to keep in mind that they do not in the end do tasks that real people actually do, unlike MT and IE systems. 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 by many people with no such subjective commitments.

4 Application Areas of Information Extraction

In section 2 we reviewed work in IE from an historical perspective, describing efforts in the area in a chronological fashion. It is also of interest, however, to view IE from the perspective of the application areas in which IE systems have been or are being deployed. This perspective should help to dispel the view, which the MUC evaluations may have unintentionally engendered, that IE is only of interest for military intelligence or financial applications, and to stimulate thinking about the range of potential applications for this growth technology.

The following list is bound to be partial; but it is indicative of the range of areas in which IE technology is already in play.

Finance The MUC-5 joint ventures scenario lead at least thirteen sites to develop IE systems for extracting details of joint ventures from newswire stories [18]. The MUC-6 management succession event scenario is also of potential interest to those working in finance [19]. The COBALT and FACILE projects [28, 29] which use IE techniques to help categorise newswire stories of relevance to stock traders also operate in this area. A number of companies have expressed interest to the authors in competitor intelligence systems that will enable them to track ventures in which their competitors are engaged, as reported in newswires.

Military intelligence The U.S. Air Force supported early research on the extraction of satellite events [58]. MUC-1 and MUC-2 focussed on hostile actions of enemy units against U.S. naval forces. MUC-3 and MUC-4 concentrated on gathering information about terrorist attacks from Latin American newsfeeds [17, 16].

Medicine Sager's early work [5] illustrated the possibility of gathering information from patient discharge summaries and radiology reports. Work by Lehnert also applied IE in a medical domain [59]. We have discussed applications of IE with local medical informatics experts and they confirm the need for applications to help in the classification of patient records and discharge summaries to assist in public health research and in medical treatment auditing.

Law The NAVILEX project aims to use IE techniques to support intelligent retrieval from legal texts [60]. It follows on from the NOMOS project which also applied 'shallow' NLP techniques to extract information from legal texts to assist in retrieval [61].

Police The POETIC project developed an IE system for extracting information about road traffic incidents from police 'command and control' incident logs [25]. The AVENTINUS project is working to build tools to assist police in criminal investigations relating to drug trafficking [31, 32].

Technology/product tracking One of the two MUC-5 extraction scenarios was microelectronics products announcements – extracting details about new microelectronic technology from the trade press [18]. Again, industrialists have expressed an interest to us in tracking commodity price changes and factors affecting these changes in the relevant newsfeeds.

Academic research Academic journals and publications are increasingly becoming available on-line and offer a prime, if challenging, source of material for IE technology. The EMPathIE project in which we are currently involved is exploring the possibility of building an Enzyme and Metabolic Pathways database using IE techniques to fill in templates about enzymes and enzyme activities from electronic versions of relevant biomolecular journals [62]. Cowie's work on wild flower guides[14] and Zarri's work on historical texts [15] are early examples of this sort of work.

Employment The TREE project aims to build a database of employment opportunities from electronic job advertisements [26, 27].

Fault Diagnosis The SINTESI project extracts information from reports of car faults [3]; the TACITUS system was also employed in analysing engine failure reports [63, 64].

Software system requirements specification NLP techniques have been used to assist in the process of deriving formal software specifications from less formal, natural language specifications. We are currently involved in research to see if this problem can be cast in the form of an IE problem, where the formal specification is viewed as a template which needs to be filled from a natural language specification, supplemented with a dialogue with the user.

Together these applications demonstrate the broad range of projects already undertaken or in progress which utilise IE technology. Clearly they represent but a tiny fraction of potential applications – which supports our claim to the importance of IE as a growth text processing technology.

5 Concluding Remarks

5.1 Challenges for the Future

We hope the foregoing discussion has illuminated the objectives of IE, the as yet brief history of this area of research, the sorts of approaches that are being used, and the areas of application which have been and are being considered. In concluding we focus on a number of central challenges facing IE in the future.

5.1.1 Higher Precision and Recall

Combined precision and recall scores for IR systems have rested in the mid-50% range for many years, and it is in this range that current IE systems also find themselves. While users of IR systems have adapted themselves to these performance levels, it is not clear that for IE applications such levels are acceptable. Clearly what is tolerable will vary from application to application. But where IE applications involve building databases over extended periods of time which subsequently form the input to further analysis, noise in the data will seriously compromise its utility. Cowie and Lehnert [4] suggest that 90% precision will be necessary for IE systems to satisfy information analysts. Current high precision scores in the MUC scenario extraction tasks are around 70%.

Improvements in both precision and recall are high priority challenges for IE systems. There are no 'magic bullets' on the horizon, but there is every reason to believe that significant progress can be made as research continues in NLP and as more lexical and grammatical resources become available.

5.1.2 User-defined IE

Currently IE systems are tailored for new applications through a two stage process which involves first defining a template for the application – identifying the entities, attributes and relations to be captured – and second modifying the lexical, grammatical and conceptual rule-bases that the IE system uses in carrying out its text processing. Both of these stages typically require the involvement of experts. The first requires a logical analysis of the information to be captured and the articulation of this analysis in a particular formalism. Given that the second stage of the customisation is highly dependent on this first stage and will require considerable effort, it is important that this stage be carried out correctly and given the current development of the technology this is only probable if the person defining the template has a good grasp of the nature and limits of IE systems.

The second stage of customisation – modifying the lexical, grammatical and conceptual rule-bases that the IE system uses in carrying out its text processing – clearly requires expert knowledge. If these rule-bases are handcrafted, then those with the knowledge to do the handcrafting – typically computational linguists or NLP experts – must perform the customisation for each new domain. If the rule-bases are not handcrafted, but acquired from corpora, then the corpora must be carefully selected, perhaps annotated, and the rule acquisition process monitored carefully.

Thus porting IE systems to new domains is a serious bottleneck for state-of-the-art systems. As a consequence, the development of IE technology that permits users to define the extraction task and then adapts to the new scenario is a major challenge: only with the development of such user-centred, adaptive systems is IE technology likely to become of utility to information gathers other than those who can afford to dedicate months of expensive customisation effort to the task.

Some progress has been made in this direction. The final MUC-6 scenario task was only given to participants one month before the evaluation in an effort to reward highly portable systems. SRA have begun developing tools to help users define templates through examples [51]. Morgan *et al.* [65] have also experimented with various techniques to allow users to customise the Lolita system for new IE tasks.

5.1.3 Integration with other Technologies

IE need not be considered a standalone technology which is of use only for applications in which a structured database is to be created from a text corpus. There are a number of other technologies with which it might be combined to yield powerful new information gathering capabilities.

Information Retrieval The TIPSTER programme from the very start conceived of IR and IE as naturally forming two stages of a coupled information gathering effort, referring to them as detection and extraction respectively. The assumption was that an initial user query would be given to an IR system which from a potentially massive document collection would detect the relevant documents to be passed on to an IE system for the more detailed and computationally intensive analysis that such systems carry out.

While this coupling was initially conceived of in the context of the massive electronic document collections being assembled by governments and other large organisations, the arrival of the WWW has made available a document collection whose size threatens to dwarf anything the TIPSTER convenors conceived of as little as five years ago.

Despite the natural complementarity of IR and IE we are not aware of much practical work which has gone on in this direction as yet. We have done some preliminary experimental work in using Web search engines to create document collections which are then processed by the LaSIE system, and are encouraged by the results [66, 67]. However much more work needs to be done in this area, and no doubt will be.

Aside from this obvious way of combining IR and IE systems, there are other possible ways in which the two technologies may be of mutual benefit. Specifically, for applications where the computational intensiveness of IE systems is not a drawback, an IE system could be used in conjunction with the indexing component of an IR system in one of a number of ways. Most obviously, the proper name recognition and classification abilities of an IE system could be harnessed to provide useful (possibly) multi-word, preclassified index terms that would enable searches for, e.g., ‘Ford’ the company, and exclude all references to persons and places named ‘Ford’. But more sophisticated indexing could be developed based on the identification of entities and relations, such as IE systems carry out. For example, remaining with the management succession scenario, one could index documents according to succession events and roles in them so that one could search for all reports mentioning persons who had resigned from CEO positions in Canadian companies in the last year. Work on using IE templates for indexing legal documents is implemented in the Navilex system [60]; work on using IE techniques to supplement traditional IR approaches to categorising and filtering news stories is being carried out in the related COBALT and FACILE projects, as mentioned above in section 2.3. Clearly there are many further potential applications of this nature.

Natural Language Generation Our example in Figure 1 showed the NL summary the LaSIE system generated from the template it had extracted. This summary was generated using very crude generation techniques. Given that much more sophisticated NL generation (NLG) capabilities now exist [68], the coupling of IE and NLG should permit more fluid, easy to read summaries to be generated from extracted templates.

Machine Translation The translation of documents may be carried out for many reasons, but if the purpose of the translation is to enable subsequent extraction of information from the text that was previously inaccessible to the information seeker because of the language barrier, then given the difficulty of translation it is worth considering ways in which the information sought could be first extracted and then translated. That is, rather than performing translation followed by extraction, it may be preferable to perform extraction in the source language followed by translation into the destination language. Such a coupling of IE and MT technologies is particularly attractive because a template, being regularised provides a much easier information source to translate than a full text.

Some work along these lines has already been carried out [69, 70] but we expect much more work to be carried out in this area in the near future. Again, given the sudden availability of multilingual on-line text afforded by the Web, information gatherers will want ways of accessing this information that avoid the overheads of large scale translation.

Data Mining IE systems produce structured data repositories which can be turned into conventional databases to be accessed with conventional database access tools such as SQL query processors. However, these databases may also be processed by data mining (DM) or knowledge discovery in database (KDD) tools which seek novel patterns in the data [71]. The significance of coupling IE with DM or KDD techniques is that this will permit hitherto unmined text resources to become the subject of extensive exploration. As one example, consider the possibilities of extracting information about commodity price changes from financial news reports, building a database of these fluctuations over some historical period and then using KDD techniques to discover correlations that might give insights into the causes of these changes. Once again, coupling IE with another technology promises powerful new techniques for gathering information from texts.

5.2 IE or not IE?

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 perform IE against 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 summarised, 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.

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 formalised 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 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 formalised 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 formalised 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 favourite movie critic likes', addressed to the right text corpus, is undoubtedly IE, 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.

Acknowledgments

Thanks to Beth Sundheim of the US Naval Command, Control, and Ocean Surveillance Center RDT&E Division (NRaD) for detailed comments during the preparation of this paper.

The University of Sheffield's LaSIE system and the GATE architecture described here were supported by the UK EPSRC (grant number GR/K25267).

References

1. WITTEN, I.H., MOFFAT, A., & BELL, T.C. *Managing Gigabytes.* New York: Van Nostrand Reinhold, 1994.

2. JACOBS, P.S., & RAU, L.F. SCISOR: Extracting information from on-line news. *Communications of the ACM*, 33(11), 1990, 88–97.

3. CIRAVEGNA, F., CAMPIA, P., & COLOGNESE, A. Knowledge Extraction from Texts by SINTESI. *In: Proceedings of the Fifteenth International Conference on Computational Linguistics (COLING-92).* 1992, 1244–1248.

4. COWIE, J., & LEHNERT, W. Information Extraction. *Communications of the ACM*, 39(1), 1996, 80–91.

5. SAGER, N. *Natural Language Information Processing.* Reading, Massachusetts: Addison-Wesley, 1981.

6. COLLIER, R. *Automatic Template Creation for Information Extraction.* Technical Report CS-96-07. Department of Computer Science, University of Sheffield, UK, September 1996.

7. SCHANK, R.C., & COLBY, M.C. *Computer Models of Thought and Language.* San Francisco: W.H. Freeman, 1973..

8. SCHANK, R.C. *Conceptual Information Processing.* Amsterdam: North-Holland, 1975.

9. SCHANK, R.C., & ABELSON, R.P. *Scripts, Plans, Goals, and Understanding.* Hillsdale, N.J.: Lawrence Erlbaum Associates, 1977.

10. SCHANK, R.C., & RIESBECK, C.K. *Inside Computer Understanding: Five Programs Plus Miniatures.* Hillsdale, N.J.: Lawrence Erlbaum Associates, 1981.

11. DEJONG, G. An Overview of the FRUMP System. *In:* LEHNERT, W., & RINGLE, M.H. (eds), *Strategies for Natural Language Processing.* Lawrence Erlbaum, 1982, 149–176.

12. LYTINEN, S.L., & GERSHMAN, A. ATRANS: Automatic processing of money transfer messages. *In: Proceedings of the Fifth National Conference on Artificial Intelligence (AAAI-86).* 1986, 1089–1093.

13. ANDERSEN, P.M., HAYES, P.J., HUETTNER, A.K., NIRENBURG, I.B., SCHMANDT, L.M., & WEINSTEIN, S.P. Automatic Extraction of Facts from Press Releases to Generate News Stories. *In: Proceedings of the Third Conference on Applied Natural Language Processing.* 1992, 170–177.

14. COWIE, J.R. Automatic Analysis of Descriptive Texts. *In: Proceedings of the ACL Conference on Applied Natural Language Processing.* 1983, 117–123.

15. ZARRI, G.P. Automatic Representation of the Semantic Relationships Corresponding to a French Surface Expression. *In: Proceedings of the ACL Conference on Applied Natural Language Processing.* 1983, 143–147.

16. *Proceedings of the Third Message Understanding Conference (MUC-3).* Morgan Kaufmann, for Defense Advanced Research Projects Agency, 1991.

17. *Proceedings of the Fourth Message Understanding Conference (MUC-4).* Morgan Kaufmann, for Defense Advanced Research Projects Agency, 1992.

18. *Proceedings of the Fifth Message Understanding Conference (MUC-5).* Morgan Kaufmann, for Advanced Research Projects Agency, 1993.

19. *Proceedings of the Sixth Message Understanding Conference (MUC-6).* Morgan Kaufmann, for Defense Advanced Research Projects Agency, 1995.

20. SUNDHEIM, B. Tipster/MUC-5 Information Extraction System Evaluation. *In: Proceedings of the Fifth Message Understanding Conference (MUC-5).* Morgan Kaufmann, 1993, 27–44.

21. VAN RIJSBERGEN, C.J. *Information Retrieval.* London: Butterworths, 1979.

22. VILAIN, M., BURGER, J., ABERDEEN, J., CONNOLLY, D., & HIRSCHMAN, L. A Model-Theoretic Coreference Scoring Scheme. *In: Proceedings of the Sixth Message Understanding Conference (MUC-6).* Morgan Kaufmann, 1995, 45–52.

23. CHINCHOR, N. The Statistical Significance of the MUC-5 Results. *In: Proceedings of the Fifth Message Understanding Conference (MUC-5).* Morgan Kaufmann, 1993, 79–83.

24. MILLER, G.A. WordNet: An on-line Lexical Database. *International Journal of Lexicography,* 3(4), 1990, 235–312.

25. EVANS, R., GAIZAUSKAS, R., CAHILL, L., WALKER, J., RICHARDSON, J., & DIXON, A. POETIC: A System for Gathering and Disseminating Traffic Information. *Journal of Natural Language Engineering,* 1(4), 1995, 363–387.

26. ELLMAN, J., SOMERS, H., NIVRE, J., & MULTARI, A. Foreign Language Information Extraction: An Application in the Employment Domain. *In: Natural Language Processing: Extracting Information for Business Needs.* Unicom Seminars Ltd., London, March 1997, 77–89.

27. *TREE: Trans European Employment.* http://www2.echo.lu/langeng/en/le1/tree/tree.html. Site visited 29/05/97.

28. ROCCA, G., SPAMPINATO, L., ZARRI, G.P., BLACK, W., & CELNIK, P. COBALT: Construction, Augmentation and Use of Knowledge bases from Natural Language Documents. *In: Proceedings of the Artificial Intelligence Conference.* May 1994.

29. BLACK, W.J. FACILE: Fine-Grained Multilingual Text Categorisation and Information Extraction. *In: Natural Language Processing: Extracting Information for Business Needs.* Unicom Seminars Ltd., London, March 1997, 119–131.

30. *FACILE: Fast and Accurate Categorisation of Information by Language Engineering.* http://www2.echo.lu/langeng/en/le1/facile/facile.html. Site visited 29/05/97.

31. THURMAIR, G. Information Extraction for Intelligence Systems. *In: Natural Language Processing: Extracting Information for Business Needs.* Unicom Seminars Ltd., London, March 1997, 135–149.

32. *AVENTINUS: Advanced Information System for Multinational Drug Enforcement.* http://www2.echo.lu/langeng/en/le1/aventinus/aventinus.html. Site visited 29/05/97.

33. *ECRAN: Extraction of Content: Research at Near-Market.* http://www2.echo.lu/langeng/en/le1/ecran/ecran.html. Site visited 29/05/97.

34. HOBBS, J.R. The Generic Information Extraction System. *In: Proceedings of the Fifth Message Understanding Conference (MUC-5).* Morgan Kaufman, 1993, 87–91.

35. GAIZAUSKAS, R.G., CUNNINGHAM, H., WILKS, Y., RODGERS, P., & HUMPHREYS, K. GATE – an Environment to Support Research and Development in Natural Language Engineering. *In: Proceedings of the 8th IEEE International Conference on Tools with Artificial Intelligence (ICTAI-96).* October 1996, 58–66.

36. CUNNINHAM, H., HUMPHREYS, K., GAIZAUSKAS, R., & WILKS, Y. Software Infrastructure for Natural Language Processing. *In: Proceedings of the Fifth Conference on Applied Natural Language Processing (ANLP-97).* Available at `http://xxx.lanl.gov/ps/9702005`. March 1997, 237–244.

37. GRISHMAN, R. *TIPSTER Architecture Design Document Version 2.2.* Technical Report. Defense Advanced Research Projects Agency. Available at `http://www.tipster.org/`. 1996.

38. HUMPHREYS, K., GAIZAUSKAS, R., CUNNINGHAM, H., & AZZAM, S. *VIE Technical Specifications.* Department of Computer Science, University of Sheffield, 1996.

39. BRILL, E. A Simple rule-based part-of-speech tagger. *In: Proceeding of the Third Conference on Applied Natural Language Processing.* 1992, 152–155.

40. MARCUS, M.P., SANTORINI, B., & MARCINKIEWICZ, M.A. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2), 1993, 313–330.

41. WAKAO, T., GAIZAUSKAS, R., & WILKS, Y. Evaluation of an Algorithm for the Recognition and Classification of Proper Names. *In: Proceedings of the 16th International Conference on Computational Linguistics (COLING96).* 1996, 418–423.

42. GAZDAR, G., & MELLISH, C. *Natural Language Processing in Prolog.* Wokingham: Addison-Wesley, 1989.

43. GAIZAUSKAS, R. *Investigations into the Grammar Underlying the Penn Treebank II.* Technical Report CS-95-25. Department of Computer Science, University of Sheffield, 1995.

44. GAIZAUSKAS, R. *XI: A Knowledge Representation Language Based on Cross-Classification and Inheritance.* Technical Report CS-95-24. Department of Computer Science, University of Sheffield. 1995.

45. GAIZAUSKAS, R., & HUMPHREYS, K. Quantative Evaluation of Coreference Algorithms in an Information Extraction System. *In:* BOTLEY, S., & MCENERY, T. (eds), *Discourse Anaphora and Anaphor Resolution.* University College London Press, 1997, (in press).

46. HOBBS, J.R. Description of the TACITUS System as Used for MUC-3. *In: Proceedings of the Third Message Understanding Conference MUC-3.* Morgan Kaufmann, 1991, 200–206.

47. HOBBS, J.R., APPELT, D., TYSON, M., BEAR, J., & ISRAEL, D. Description of the FASTUS System as Used for MUC-4. *In: Proceedings of the Fourth Message Understanding Conference MUC-4.* Morgan Kaufmann, 1992, 268–275.

48. APPELT, D.E., HOBBS, J.R., BEAR, J., ISRAEL, D., KAMEYAMA, M., & TYSON, M. Description of the JV-FASTUS System as Used for MUC-5. *In: Proceedings of the Fourth Message Understanding Conference MUC-5.* Morgan Kaufmann, 1993, 221–235.

49. APPELT, D., HOBBS, J., BEAR, J., ISRAEL, D., KAMEYAMA, M., KEHLER, A., MARTIN, D., MYERS, K., & TYSON, M. SRI International FASTUS system: MUC-6 Test Results and Analysis. *In: Proceedings of the Sixth Message Understanding Conference (MUC-6).* Morgan Kaufmann, 1995, 237–248.

50. ABERDEEN, J., BURGER, J., DAY, D., HIRSCHMAN, L., ROBINSON, P., & VILAIN, M. MITRE: Description of the Alembic System Used for MUC-6. *In: Proceedings of the Sixth Message Understanding Conference (MUC-6).* Morgan Kaufmann, 1995, 141–156.

51. KRUPKA, G.R. Description of the SRA System as used for MUC-6. *In: Proceedings of the Fourth Message Understanding Conference (MUC-6).* Morgan Kaufmann, 1995, 221–236.

52. GRISHMAN, R. *TIPSTER Architecture Design Document Version 1.52 (Tinman Architecture).* Technical Report. Department of Computer Science, New York University. Available at `http://www.cs.nyu.edu/tipster`. 1995.

53. GRISHMAN, R., & STERLING, J. Description of the Proteus System as Used for MUC-5. *In: Proceedings of the Fifth Message Understanding Conference (MUC-5).* Morgan Kaufmann, 1993, 181–194.

54. LIN, D. Description of the PIE System as used for MUC-6. *In: Proceedings of the Sixth Message Understanding Conference (MUC-6).* San Francisco: Morgan Kaufmann, 1995, 113–126.

55. WEISCHEDEL, R. Description of the PLUM System as used for MUC-6. *In: Proceedings of the Sixth Message Understanding Conference (MUC-6).* San Francisco: Morgan Kaufmann, 1995, 55–70.

56. WILKS, Y., GUTHRIE, L., & SLATOR, B. *Electric Words.* Cambridge, MA: MIT Press, 1996.

57. GRISHMAN, R., & SUNDHEIM, B. Message Understanding Conference - 6: A Brief History. *In: Proceedings of the 16th International Conference on Computational Linguistics.* June 1996, 466–471.

58. MONTGOMERY, C. Distinguishing Fact from Opinion and Events from Meta-Events. *In: Proceedings of the ACL Conference on Applied Natural Language Processing.* 1983.

59. LEHNERT, W., SODERLAND, S., ARONOW, D., FENG, F., & SHMUELI, A Inductive Text Classification for Medical Applications. *Journal for Experimental and Theoretical Artificial Intelligence*, 7(1), 1994, 49–80.

60. PIETROSANTI, E., & GRAZIADIO, B. Artificial Intelligence and Legal Text Management: Tools and Techniques for Intelligent Document Processing and Retrieval. *In: Natural Language Processing: Extracting Information for Business Needs.* Unicom Seminars Ltd., London, March 1997, 277–291.

61. GIANETTI, A., DASSOVICH, P., MARCHIGNOLI, G., MUSSETTO, P., PIETROSANTI, E., AZZAM, S., CELNIK, P., BILON, J., FORTIER, V., & PIRES, F. NOMOS: Knowledge Acquisition for Normative Reasoning Systems. *In:* STEELS, L., & LEPAPE, B. (eds), *Enhancing the Knowledge Engineering Process: Contributions from ESPRIT.* Elsevier Science Publications, 1992.

62. *EMPathIE: Enzyme and Metabolic Path Information Extraction.*
http://www.dcs.shef.ac.uk/research/groups/nlp/funded/empathie.html.
Site visited 29/05/97.

63. HOBBS, J.R. The TACITUS Project. *Computational Linguistics*, 12(3), 1986, 220–222.

64. HOBBS, J.R., STICKEL, M.E., APPELT, D.E., & MARTIN, P. Interpretation as Abduction. *Artificial Intelligence*, 63, 1993, 69–142.

65. MORGAN, R.G. *An architecture for user defined information extraction.* Technical Report 8/96. Department of Computer Science, University of Durham, 1996.

66. ROBERTSON, A.M., & GAIZAUSKAS, R.. On the Marriage of Information Retrieval and Information Extraction. *In:* FURNER, J., & HARPER, D.J. (eds), *Proceedings of the 19th Annual BCS-IRSG Colloquium on IR Research.* 1997, 60–67.

67. GAIZAUSKAS, R., & ROBERTSON, A.M. Coupling Information Retrieval and Information Extraction: A New Text Technology for Gathering Information from the Web. *In: Proceedings of the 5th Computed-Assisted Information Searching on Internet Conference (RIAO'97).* 1997, 356–370.

68. UZKOREIT, H. Language Generation. *In: Survey of the State of the Art in Human Language Technologies.* http://www.cse.ogi.edu/CLSU/HLTsurvey/HLTsurvey.html: Centre for Spoken Language Understanding, Oregon Graduate Institute, 1997.

69. KAMEYAMA, M. Information Extraction across Linguistic Barriers. *In: AAAI Spring Symposium on Cross-Language Text and Speech Retrieval.* March 1997.

70. AZZAM, S., HUMPHREYS, K., GAIZAUSKAS, R., CUNNINGHAM, H., & WILKS, Y. Using a Language Independent Domain Model for Multilingual Information Extraction. *In:* SPRYRODOPOULOS, C. (ed), *Proceedings of the IJCAI-97 Workshop on Multilinguality in the Software Industry: the AI Contribution (MULSAIC-97).* 1997, (in press).

71. FAYYAD, U., PIATETSKY-SHAPIRO, G., & SMYTH, P. From Data Mining to Knowledge Discovery in Databases. *AI Magazine.* 1996, 37–54.