SUSSEX UNIVERSITY: DESCRIPTION OF THE SUSSEX SYSTEM USED FOR MUC-5

Robert Gaizauskas, Lynne Cahill & Roger Evans
Cognitive and Computing Sciences,
University of Sussex,
Brighton, UK

robertg@cogs.susx.ac.uk
lynneca@cogs.susx.ac.uk
rogere@cogs.susx.ac.uk

INTRODUCTION

This paper describes the system used for the University of Sussex team's participation in the MUC-5 message understanding trials. What is described below is the result of 12 person-months of intensive effort over six months to adapt a pre-existing system, designed with very different objectives and application in mind, to the MUC-5 English Joint Ventures task. This task, starting from cold, is colossal: the overhead of understanding the task, the training data, the scoring, the background resources, of developing a suitable harness for the system, not to mention sorting out contractual arrangements, leaves little time for even basic porting – actual development tailored to the task was a very remote prospect. So, despite the quirks and failings exposed by the discussion below of the 'walkthrough' example, we are pleased with our system's performance, and believe the effort to have been a worthwhile part of our ongoing research.

HISTORY

The system used for MUC-5 is a modified version of a system developed at Sussex as part of the TIC ('Traffic Information Collator') and subsequently POETIC ('Portable Extendable Traffic Information Collator') projects, in association with Racal Research Ltd., the Automobile Association, and National Transcommunications Ltd., and part-funded by the UK Science and Engineering Research Council and Department of Trade and Industry.

POETIC is a prototype software system which monitors police reports of traffic incidents and automatically generates 'traffic bulletins' for motorists. POETIC's inputs are police incident reports entered as text into a police logging computer – database entries containing free format text fields which describe the incident in telegrammatic, jargon-laden English, as well as some information in fixed field format. The system identifies reports about traffic incidents, and uses them to build up a picture of the key features of an incident. From this information it makes judgements about the effect, seriousness, duration etc. of each incident, formulates suitable advisory messages, and coordinates delivery of the messages to the affected motorists. Further details of POETIC can be found in [1], [11], [14], [5].

The initial stages of the POETIC system constitute a message understanding system very similar in function to the systems participating in the MUC evaluations: mapping from narrow-domain free text input to a semantic representation of key pieces of information. Furthermore the 'Portable' in POETIC refers to domain portability – the system is structured to make porting between domains relatively straightforward, albeit only in the narrow sense of different police sublanguage domains all within the same overall domain of traffic reports. But although POETIC is a fairly mature system (more than 16 person-years of development effort over the past 8 years, of

which we estimate two thirds has been on the message understanding component), it has to date remained rather isolated in its own field of traffic management systems, in which there is little similar work. In particular, the message understanding component has been evaluated primarily only relative to the requirements of the rest of the POETIC system, rather than in an independent context.

Thus our objectives in participating in MUC-5 were threefold: firstly, to see how readily POETIC's message understanding component (designed with a particular type of domain in mind) could be adapted to a radically different topic domain, secondly to get a more objective view of how well the system performs, and how it relates to other approaches, and finally to take a further step towards the larger goal of developing a generic message understanding system – portable to a much wider range of (still narrow) domains.

SYSTEM DESCRIPTION

Key Design Features

The key strengths and novelties of our approach to message understanding are:

- the declarative representation of all its main knowledge bases (lexicon, grammar, domain model)
- support for extendability and domain-portability, through identification and localisation of domain-specific information
- the 'interesting corner' island parsing technique, which allows grammar coverage to focus on domain-relevant phenomena
- the coupling of robust but overfragmented partial analysis of input sentences with a semantic modeller which utilises domain knowledge to construct a coherent interpretation of extended texts
- the application and integration in a practical information extraction system of theoretically well-founded techniques – bidirectional chart parsing, unification-based grammar, compositional semantics, inheritance-based representation languages.

Architecture

The architecture of the Sussex system used for MUC is shown in Figure 1.

The Pre-processor The Pre-processor takes an input document[1] and massages it into a form more easily processed by the rest of the system. Its key function is to break the document up into 'fixed field' sentences (such as document number, date and source), and free text sentences. To do this, it interprets SGML tags where present, looks for sentence boundaries in conventional ways, and tokenises the resulting text – identifying numbers and punctuation symbols and separating them from alphabetic text etc.. It also attempts to locate quoted speech and discards it – examination of the training corpora revealed that very rarely was there useful information in such text which was not also elsewhere, and far more often it proved quite troublesome to process. The result is a sequence of distinct sentences, either fixed field or free text.

Fixed field sentences bypass lexical and syntactic processing: they are converted directly into fake semantic parses and passed to the Discourse Interpreter. Free text sentences are passed to the lexical analyser and then the parser.

The Lexical Analyser The Lexical Analyser takes pre-processed sentences and performs a sequence of tests, looking for lexical phrases and lexical entries for individual words. It is primarily here that background resources (company names, locations, personal names etc.) are employed. Broadly speaking, this process splits into three phases: looking for lexical phrases with reliable indicators, looking in the main lexicon for known words, and then looking for lexical phrases with less reliable indicators. The aim here, and indeed throughout the

[1] Or, in test mode, a corpus, which it splits up into separate documents.

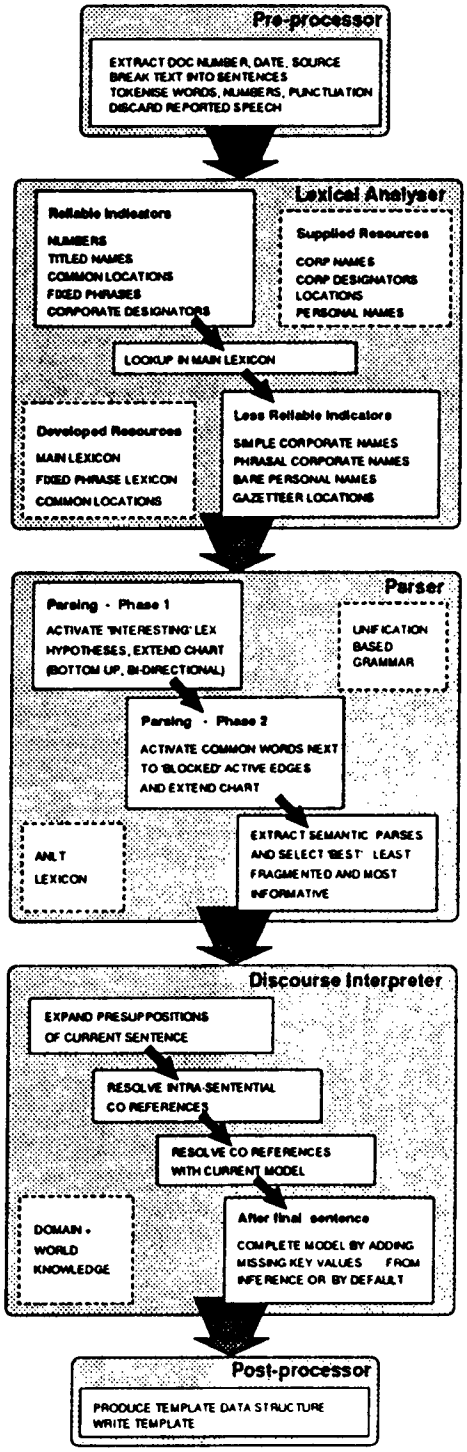

Figure 1: System Architecture

lexical and syntactic analysis, is to minimise the number of hypotheses so that fairly conventional parsing with a fairly conventional grammar remains computationally feasible. To this end the search is ordered so that more probable interpretations of a word are sought first, and if found will often preclude the hypothesis of less probable alternatives.

The following reliable phrasal indicators are detected:

Numbers some of which may have alternative interpetations as years, or be followed by an ordinal marker (*st, nd, rd etc.*).

Titled names where the titles come from a fixed set, and special-purpose name detection routines locate the entire name phrase. Titles can be prefixed or postfixed and most become incorporated in the resulting lexical phrase. A few, however, such as *president* and *chairman* have their own entries in the main lexicon, and so trigger a name phrase but remain independent of it so that the appropriate semantic composition takes place during parsing.

Common locations gleaned from the training corpus. The supplied gazetteer is far too unwieldy to use directly and the timescale did not permit much experimentation with it. Instead we compiled a list of the most frequent locations (cities, countries regions – just over 300 in all) actually occurring in the training corpus and this is the data used here. Other locations may be picked up in the third phase of lexical processing – see below.

Fixed phrases occurring frequently in the domain, such as *joint venture, trading house, stock exchange*, mostly derived from a digram analysis of the training corpus. Also, common English fixed phrases such as *as well as*.

Corporate designators whose presence triggers special-purpose company name detection code. Company names identified in this way are remembered so that they will be recognised later in the document even without a corporate designator[2].

The main domain-specific lexicon is written in the inheritance-based lexical representation language DATR [10], [9]. This contains the root forms of significant domain words (e.g. *company, own*) and the most common English words (e.g. *the, and*) found in the training corpus – about 600 entries in all, although the level of semantic detail is as yet far from uniform. Before looking up a word in this lexicon, a simple morphological analysis is performed: a set of rules is used to detect standard suffixes, resulting in a root + suffix analysis for each word. The root indexes a lexical entry (or entries, for ambiguous forms) while the suffix information is used to fill in feature details. Further details of this mechanism can be found in [4], [3].

The following less reliable phrasal indicators are only accepted cautiously:

Simple corporate names from the supplied resources. One-word company names are sought *after* the main lexicon lookup. Thus a normal domain interpretation of a word is preferred over any possible company name intepretation. However the presence of a following corporate designator will generally force a company interpretation when it is encountered.

Phrasal corporate names from the supplied resources. A phrasal name is split into an essential (contiguous) core, plus peripheral optional parts. Phrases are indexed on the first word of the core, and if the whole core is located, then optional parts will also be incorporated into the lexical phrase if present. In principle, this might be a very reliable match but both the automatic identification of appropriate cores in the corpus, and the heuristics for matching optional words are still rather rudimentary.

Bare personal names using the names resources and a certain amount of guesswork about unknown words.

[2]This is in fact the only 'global state' maintained by the lexical analyser or the parser – all other lexical and syntactic processing looks only at the current sentence.

Gazetteer locations from the supplied gazetteer – using this directly leads to many surprising, and generally spurious hits, and much irrelevant ambiguity even in sensible cases, so it is employed very much as a last resort at present.

Note that the main lexicon contains only about 600 words, and since many words in the input will neither be in this lexicon nor be part of recognisable lexical phrases, many words will be given no lexical analysis whatsoever by the Lexical Analyser. Avoiding complete initial lexical analysis is one of the features of our approach; limited further lexical analysis is done during parsing, as described in the next section.

The Parser The Parser takes the initial lexical hypotheses for a single sentence produced by the lexical analyser, and attempts to build larger syntactic structures (ideally, but rarely, a complete parse). From these it builds semantic representations. The lexical entries returned (for single words and larger phrases) have three basic components: a backbone category, some feature/value specifications, and a semantic expression. The grammar rules similarly consist of backbone categories and feature specifications, plus semantic composition rules describing how the semantics of the mother category is constructed from that of the daughters.

The backbone categories are parsed using a chart parser in a conventional fashion: the fact that they are atomic allows greater efficiency in rule indexing and primary category matching. Once backbone categories have been matched, the corresponding feature sets are unified together. The most important role of unification is to enforce domain-specific type restrictions which play an important role in the parsing process. The parser has a sophisticated semantic type system, which uses term unification to implement type-compatibility (see [13] for further details).

As well as the restrictive effects of the type system in the grammar, the other main aspect which distinguishes the parser from standard chart parsers is the use of 'interesting-corner' parsing (see [13]). A grammar rule is only triggered if its indexing category is currently 'interesting'. What constitutes an interesting category is controlled dynamically by the parser and grammar and is quite domain-specific. Initially, just a few classes of lexical categories are deemed interesting (such as company names and joint venture phrases), and so only parsing from such words or phrases is ever initiated. This strategy means that the parser has to be bidirectional, but it does restrict it to analysing phrases which have some interesting content. The 'interestingness' can spread to features which are explicitly sought by a rule which has been triggered.

The parser also includes heuristics for handling common words not in the main domain lexicon. Once the chart has been extended as far as it can be on the basis of the initial hypotheses (which included only domain specific words and very common English words), the words which are 'blocking' active chart edges are examined. These words are sought in the 'ANLT' lexicon, a general lexicon of English developed under the 'Alvey Natural Language Tools' Project [2] which contains about 7000 root forms and associated with each a syntactic category. If the 'blocking' words are in the ANLT lexicon and are of the right category to extend the active edge, then they are added at this point, and parsing proceeds again. In this way, the presence of the odd unknown word within interesting text does not cause a problem, but the parser is not weighed down with attempting to process all the common words in the sentence.

The parser constructs semantic representations – expressions of lambda calculus over predicate logic – as a post-process on completed syntactic parse trees. The grammar rules specify compositionally how the semantics of complex linguistic categories are constructed out of the semantics of component categories. At the highest level, in a parse tree of a complete message, all lambda variables disappear and the parser produces an expression of predicate calculus as its interpretation of the 'meaning' of the message.

The Discourse Interpreter The Discourse Interpreter integrates semantic information from each successive parsed sentence into the current semantic model. This component draws on a number of previous approaches, such as auxiliary role fillers [7], scripts [15], interpretation as abduction [12] and world models [8], [6].

The first semantic task is to add any facts presupposed by the new information, to give us a more explicit picture of the intended meaning. So for example, a 'joint venture' event presupposes at least two (distinct) joint venture participants and an activity in which the participants will engage, but this information may not be explicitly present in the input.

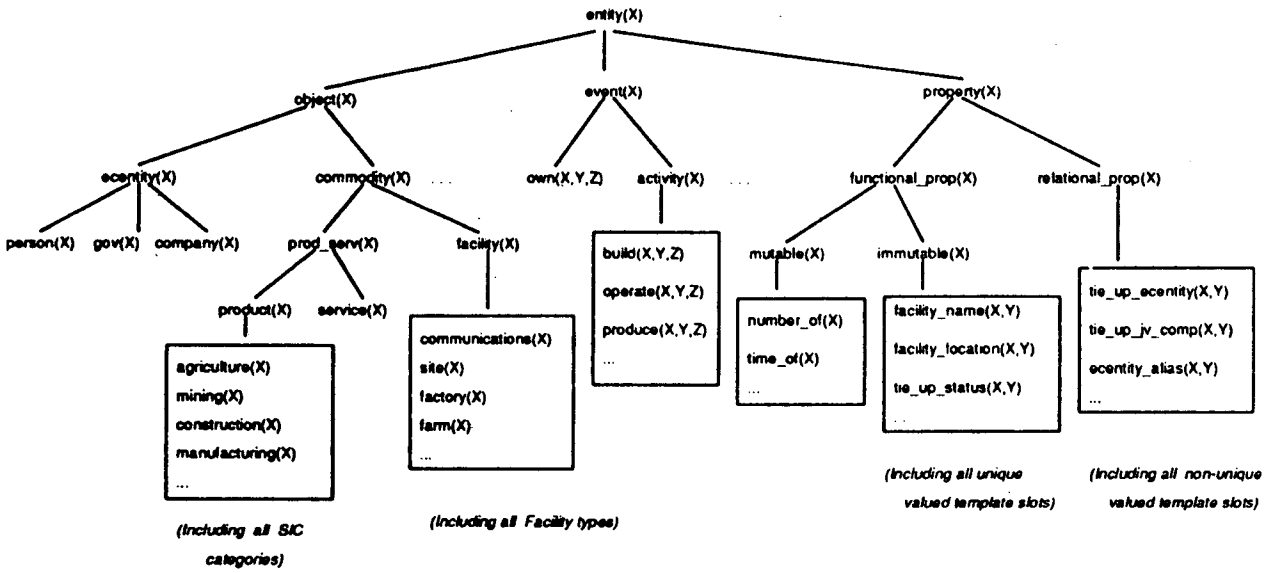

Figure 2: Fragment of EJV Ontology used in MUC-5

The second semantic task is co-reference resolution – the same entity may be referenced throughout the text in different ways, perhaps even implicitly (as in ellipsis), and this identity of reference needs to be established. To do this a data structure is employed which locates all entity types in a hierarchical type system or ontology, a fragment of which is shown in figure 2.

Each node in the ontology has properties associated with it, either defined explicitly or inherited from above. Such properties are themselves nodes in the ontology. Co-reference resolution is constrained to respect the type system (inconsistent 'types' may not co-refer) and the property specifications (some properties are unique-valued, others are not). Apart from this it is unconstrained, except that we attempt resolution *within* a parse before attempting resolution between a parse and the existing model. This is a legacy of the traffic domain, where multiple entities of the same type were rare, but as we shall see in the walkthrough, this means that inappropriate coreferences do sometimes occur.

This component also has a more subtle role in the overall architecture of the system. In order to parse robustly, the grammar has rather patchy, oversimplified coverage in 'uninteresting' areas. This means that the resultant analyses tend to be overfragmented. Part of the role of the co-reference resolver is to pull such fragments back together at the semantic level. This technique is an important factor in the overall effectiveness of our approach, allowing clearer, more conventional expression of the grammatical knowledge encoded, not encumbered by complicated support for robustness issues.

The final stage of semantic analysis, after all the sentences have been processed, is to provide values for a number of key domain features. The system has a notion of the certain features which must be present in any model, and if values for them have not been provided explicitly from the text, it will attempt to provide them itself. This is partly achieved through inference from values which *have* been provided, so that, for example, the description of an industry can be inferred from the description of a facility - mining happens at mines, manufacturing at plants etc.. When all else fails, default values are provided where appropriate. Many of these defaults are derived from statistical analysis of the training corpus and templates – for example, in the WSJ, MEAD and PROMT development template sets, ecentity-type[3] is COMPANY in 92% of cases, tie-up-status is EXISTING

[3] Throughout this paper we use the term 'ecentity' (economic entity) to denote the MUC-5 template ENTITY object – the term 'entity' being already in use as the top of our own ontology.

in 79% of cases.

The Post-processor The Post-processor simply takes the model built by the Discourse Interpreter, turns it into a template-shaped data structure and then writes it out in an appropriate format. In doing this, it traverses the template structure top down, so that any entities in the model which are not linked to the top are simply ignored.

THE WALKTHROUGH EXAMPLE

In this section we discuss the system's performance on the standard 'walkthrough' example, making reference to the more complete picture provided in the appendix. This example gave the pre-processor very little trouble, having no difficult sentence breaks or reported speech. So we begin by listing the input sentences *after* pre-processing:

1. DATE '241189'
2. SOURCE 'Jiji Press Ltd.'
3. NEWCONTEXT 1
4. BRIDGESTONE SPORTS CO . SAID FRIDAY IT HAS SET UP A JOINT VENTURE IN TAIWAN WITH A LOCAL CONCERN AND A JAPANESE TRADING HOUSE TO PRODUCE GOLF CLUBS TO BE SHIPPED TO JAPAN
5. THE JOINT VENTURE , BRIDGESTONE SPORTS TAIWAN CO . , CAPITALIZED AT 20 MILLION NEW TAIWAN DOLLARS , WILL START PRODUCTION IN JANUARY 1990 WITH PRODUCTION OF 20000 IRON AND \" METAL WOOD \" CLUBS A MONTH
6. THE MONTHLY OUTPUT WILL BE LATER RAISED TO 50000 UNITS , BRIDGESTON SPORTS OFFICIALS SAID
7. THE NEW COMPANY , BASED IN KAOHSIUNG , SOUTHERN TAIWAN , IS OWNED 75 PCT BY BRIDGESTONE SPORTS , 15 PCT BY UNION PRECISION CASTING CO . OF TAIWAN AND THE REMAINDER BY TAGA CO . , A COMPANY ACTIVE IN TRADING WITH TAIWAN , THE OFFICIALS SAID
8. BRIDGESTONE SPORTS HAS SO FAR BEEN ENTRUSTING PRODUCTION OF GOLF CLUB PARTS WITH UNION PRECISION CASTING AND OTHER TAIWAN COMPANIES
9. WITH THE ESTABLISHMENT OF THE TAIWAN UNIT , THE JAPANESE SPORTS GOODS MAKER PLANS TO INCREASE PRODUCTION OF LUXURY CLUBS IN JAPAN
10. CASE ALLUPPER

Of these, sentences 1, 2, 3 and 10 are all 'fixed field' sentences and so not parsed syntactically. 3 perhaps deserves a comment: it indicates to the Discourse Interpreter that we are beginning a new JV text (within a single document). This is of significance when handling documents which summarise several separate stories, using -- to introduce each new one (such as document 0142 of the final run). In such cases, each -- generates a NEWCONTEXT sentence.

Sentence 4 actually makes things start happening. Lexical analysis identifies BRIDGESTONE as the company BRIDGESTONE CORPORATION | JAPAN in the supplied resources database (phrasal company matching with BRIDGESTONE as core and an optional CORPORATION, which is not present in this case), and CO as a company (because all corporate designator abbreviations get mapped to their full forms – useful in some contexts, but clearly not all!). SPORT is ignored as a common word, but had the example been mixed-case, and SPORT capitalised, a single company name spanning these three words would have resulted. JOINT VENTURE and TRADING HOUSE are both identified as fixed phrases, and most other words given reasonable analyses. Sadly, GOLF CLUBS does not appear in our still incomplete list of product services. If we changed the input text to talk about 'gloves', which our system does know about, then the line PRODUCT/SERVICE: (51 "glove") would appear in the template

The parser centers its activity on interesting lexical items, here, the companies, locations and the joint venture, and produces five fragments on this sentence as follows:

1. BRIDGESTONE

```
[company(e1), dbminfo(e1,[[value,'BRIDGESTONE CORPORATION | JAPAN '],
                          [orig_text,'BRIDGESTONE']])]
```

2. CO

```
[company(e2), numberof(e2, =, 1)]
```

3. A JOINT VENTURE IN TAIWAN

```
[tie_up(e3), numberof(e4, =, 1), location(e4), at(e3, e4),
 dbminfo(e4,[[value,'Taiwan (COUNTRY)'],[type,'COUNTRY'],
             [orig_text,'TAIWAN']])]
```

4. A LOCAL CONCERN AND A JAPANESE TRADING HOUSE

```
[ecentity(e5), numberof(e5, =, 1), other_ecentity(e6),
 numberof(e6, =, 1), nationality(e6, japan)]
```

5. JAPAN

```
[location(e7), dbminfo(e7,[[value,'Japan (COUNTRY)'],
                           [type,'COUNTRY'],[orig_text,'JAPAN']])]
```

Each of these fragments is a list of logical predicates about entities $e1$, $e2$ etc.. The predicate 'dbminfo' wraps up information obtained from the supplied resources (interfaced via the DBM database library package). Notice in fragment 4 that the common word **LOCAL** has not blocked the successful detection of the full noun phrase, despite not being present in the main lexicon.

On receiving these parses, the Discourse Interpreter first attaches an instance node to the ontology for each entity referred to in the parses ($e1$, $e2$ etc.). These instance nodes are attached immediately below the most specific object or event type node by which the entity is identified in the parses. Each predicate in the parses which is a property (determined by reference to the ontology – see figure 2) is then added to the property list of each of the entity instances which occur as its arguments. Once this mapping of representations has been achieved, we end up with an instance model containing four ecentity objects, a tie_up object and a couple of location objects. The next step is to expand presuppositions, and in this case only the tie_up entity has any presuppositions: that there are two tie_up_ecentities and a tie_up_activity. This latter has its own presuppositions that it is at an activity_site and associated with an industry. So new objects are created for all these also, and we are ready to start co-reference resolution, the main task.

Since this is the first sentence of any significance, only intra-sentential co-reference resolution occurs. The company **BRIDGESTONE** unifies with **CO** (hence 'correcting' the missed common word **SPORT** to some extent) since the latter is completely generic and will unify with any company. However, for the same reason it also unifies with **LOCAL CONCERN**, incorrectly. It also unifies with one of the posited ecentities in the tie_up. It does not unify with **TRADING HOUSE** since the latter is *not* a company – it has the incompatible type 'other_ecentity'[4], nor with the second tie_up ecentity, which is constrained to be distinct from the first, leaving these two free to unify together. The resulting model looks like this:

```
z2 <-- company(_)
   Props: [numberof(z2, =, 1), distinct(z2, z5),ecentity_type(z2, COMPANY),
           dbminfo(z2,[[value,'BRIDGESTONE CORPORATION | JAPAN'],
```

[4] This is a bug in the ontology – **TRADING HOUSE** should have been classified as a subtype of 'company', rather than as an 'other_ecentity'.

```
                [orig_text,'BRIDGESTONE']]),
            ecentity_core_name(z2,'BRIDGESTONE CORPORATION | JAPAN'),
            tie_up_ecentity(z3, z2), ecentity_name(z2, 'BRIDGESTONE')]
z3 <-- tie_up(_)
   Props: [tie_up_ecentity(z3,z5),tie_up_activity(z3,z7),numberof(z3,=,1),
           tie_up_ecentity(z3,z2),at(z3,z4)]
z4 <-- location(_)
   Props: [dbminfo(z4,[[value,'Taiwan (COUNTRY)'],
                [type,COUNTRY],[orig_text,'TAIWAN']]), at(z3, z4)]
z5 <-- other_ecentity(_)
   Props: [numberof(z5, =, 1), nationality(z5, japan),distinct(z2, z5),
           tie_up_ecentity(z3,z5),ecentity_nationality(z5,'japan (COUNTRY)')]
z6 <-- location(_)
   Props: [dbminfo(z6,
           [[value,'Japan (COUNTRY)'],
            [type,COUNTRY],
            [orig_text,'JAPAN']])]
z7 <-- activity(_)
   Props: [activity_activity_site(z7,z9),
           activity_industry(z7,z8),
           tie_up_activity(z3,z7)]
z8 <-- industry(_)
   Props: [industry_product_service(z8,z10),
           activity_industry(z7,z8)]
z9 <-- activity_site(_)
   Props: [activity_activity_site(z7,z9)]
z10 <-- product_service(_)
   Props: [industry_product_service(z8,z10)]
```

$z3$ is the tie_up involving $z1$ (BRIDGESTONE/CO/LOCAL CONCERN) and $z5$ (TRADING HOUSE). $z6$ is a floating location and $z7$ to $z10$ are as yet unknown parts of the tie_up activity.

So how well have we done on this sentence? We have found a company called BRIDGESTONE, although probably not quite the right one (in mixed case we would have got it, however). We have correctly identified it and the trading house as partaking in a joint venture, but got a bit confused about the local concern. We haven't managed to get anything about the activity, primarily because we don't know anything about golf clubs.

Turning now a little more briefly to sentence 5, again the lexical analyser falls foul of SPORTS, this time returning BRIDGESTONE as before and a company called TAIWAN CO, (and once again, in mixed case it would have got it right). The presence of TAIWAN blocks the parsing of 20 MILLION NEW TAIWAN DOLLARS – a common word here would have been successfully bridged, as would NEW. And of course, if we cannot get GOLF CLUBS. it is not surprising that we do not get IRON AND " METAL WOOD " CLUBS either. So the parse ends up rather piecemeal.

Nevertheless the discourse interpreter takes the joint venture, hypothesises participating entities and so forth, and starts resolving co-references. Within the parse for the sentence alone, there are two companies (BRIDGESTONE and the incorrectly identified TAIWAN CO) and the tie_up is looking for two companies, so they get unified (the parser having missed the apposition which might have forced at least BRIDGESTONE to be the tie_up_activity). This time there is also co-reference resolution with the existing model; in fact, the entire tie_up complex gets unified. This results in a tie_up with now three participating companies – BRIDGESTONE, TAIWAN CO and JAPANESE TRADING HOUSE. (Notice that the number of tie_up_ecentities is not constrained – it is only a presupposition that there are two.)

Sentence 6 has almost no effect: nothing triggers company name recognition for BRIDGESTON, and not even mixed case input would help us here. We do have a spelling corrector which we experimented with in the POETIC project, and which would have corrected this type of error, but its computational expense was found to be out of proportion to its usefulness. In the present domain it was decided that this was even more the case, since spelling

errors in this type of input text are actually extremely rare.

In sentence 7, the system picks out some vestige of most of the companies (THE NEW COMPANY, CASTING CO, BRIDGESTONE, TAGA CO) and as a bonus A COMPANY and yet again would have done much better in mixed case. However it did not appreciate the significance of NEW, since its entry in the lexicon does not currently have any semantic content, and so got very little of the structure right. THE NEW COMPANY got unified with BRIDGESTONE, and the net effect was simply the addition of two floating companies TAGA and A to the model. The company associated with CASTING doesn't make it out of the parser – clearly it only occurred in a less favoured parse. The parser chooses between parses initially on grounds of the coverage of each parse, and then subsequently on the basis of a "confidence" measure which has not been tuned to the MUC-5 task. This is in theory a measure of the confidence of each parse calculated as a function of the semantic predicates used and the number of entities to which they refer. Since this has not been perfected, it is possible for parses which are actually 'better' to be discarded in favour of one which scores higher.

The processing of sentence 8 is also superficial, with only one noteworthy feature: CASTING is identified as a company, despite being a common word without any company cues, because it has been seen before.

Sentence 9 has the distinction of having no effect whatsoever on the model, since all the information extracted was already present.

Once all the sentences have been processed, the model is completed, by filling in key values by inference or from defaults. In this case, an ecentity_relationship and the industry_type are added, resulting in the final model shown in the appendix. From this model, the following template is produced:

```
<TEMPLATE-0592-1> :=
    DOC NR: 0592
    DOC DATE: 241189
    DOCUMENT SOURCE: "Jiji Press Ltd."
    CONTENT: <TIE_UP_RELATIONSHIP-0592-1>
<TIE_UP_RELATIONSHIP-0592-1> :=
    TIE-UP STATUS: EXISTING
    ENTITY: <ENTITY-0592-1>
            <ENTITY-0592-3>
            <ENTITY-0592-2>
    ACTIVITY: <ACTIVITY-0592-1>
<ENTITY-0592-1> :=
    NATIONALITY: Japan (COUNTRY)
    TYPE: COMPANY
    ENTITY RELATIONSHIP: <ENTITY RELATIONSHIP-0592-1>
<ENTITY-0592-2> :=
    NAME: BRIDGESTONE
    TYPE: COMPANY
    ENTITY RELATIONSHIP: <ENTITY RELATIONSHIP-0592-1>
<ENTITY-0592-3> :=
    NAME: TAIWAN CO
    TYPE: COMPANY
    ENTITY RELATIONSHIP: <ENTITY RELATIONSHIP-0592-1>
<INDUSTRY-0592-1> :=
    INDUSTRY-TYPE: PRODUCTION
<ENTITY_RELATIONSHIP-0592-1> :=
    ENTITY1: <ENTITY-0592-2>
             <ENTITY-0592-3>
             <ENTITY-0592-1>
    REL OF ENTITY2 TO ENTITY1: PARTNER
    STATUS: CURRENT
<ACTIVITY-0592-1> :=
    INDUSTRY: <INDUSTRY-0592-1>
```

ERROR-BASED METRIC

ALL OBJECTS:

ERR	UND	OVG	SUB
78	59	25	38

RICHNESS-NORMALIZED ERROR:

Min-err	Max-err
0.882	0.9062

RECALL AND PRECISION

REC	PRE	P&R F-Measure
25	46	32.54

Table 1:Full Template Scores

OBSERVATIONS

The walkthrough example does not show our system off at its best, for a number of reasons. The first is the lack of mixed case. The system's preference for mixed case is more than just naive coding: the only way to deal with caseless data is to generate more lexical hypotheses – a situation which we are keen to avoid. In any case, while we would have got more accurate company names with mixed case, it is not clear that many of the structural problems would go away. To fix those, one needs to look at the parser, or particularly the grammar.

In POETIC, the grammar is remarkably stable across police sublanguages, presumably because most of the sublanguage variation is lexical rather than grammatical. However, this grammar is less appropriate for the Joint Ventures domain, where the language is much closer to standard English. Hence a fair amount of development work is required to achieve a proper port, and of course, only a fragment of this work has been done so far. Many of our problem areas are the standard ones – apposition, coordination etc. – but it would be interesting to see whether our architecture offers more pragmatic solutions. If the parser can be persuaded to pass on just a little more grammatical information, it might be possible for the discourse interpreter to make the right associations more robustly. This is already happening to some extent: prepositions often indicate relationships between noun phrases that are difficult to capture syntactically. In the grammar, such links are reported as being simply an abstract relation between entities, and it is up to the discourse interpreter, with its far richer context, to decide what more specific relationship is appropriate.

A further weakness that the walkthrough example shows up well is our inability to deal with many objects of the same type, notably companies. In the traffic domain, it was rarely important to distinguish entities beyond their type, and so the co-reference resolution strategies used are rather eager and arbitrary. Finer control, and perhaps even backtracking, is clearly needed here.

Finally, a few words on evaluation. Table 1 gives the results for the final evaluation run, scoring against the full template[5]. Table 2 gives the results when scored against the reduced template that was used in the dry run test two months before the final run. These figures are noticably better and arguably a better assessment of the system, because the reduced template corresponds more closely to the system's abilities. The reason for this is that the full template included a number of objects and slots not present in the reduced template, which our system *never* attempted to fill – we simply did not have time to develop, any code to support them. These, of course, count as errors in the full template score, but are ignored in the reduced template score.

[5]It is interesting to note here that the walkthrough example alone gave very similar figures, and so is fairly average for our system.

ERROR-BASED METRIC

ALL OBJECTS:

ERR	UND	OVG	SUB
74	51	23	38

RICHNESS-NORMALIZED ERROR:

Min-err	Max-err
0.5311	0.5457

RECALL AND PRECISION

REC	PRE	P&R F-Measure
30	48	37.14

Table 2: Reduced Template Scores

ACKNOWLEDGEMENTS

The research described in this paper would not have been possible but for the generous support of Integral Solutions Ltd., Racal Research Ltd., Sussex University (COGS) and the Advanced Research Projects Agency, Software and Intelligent Systems Technology Office (contract no. N66001-90-D-0192, subcontract 19-940067-31 to SAIC). Evans is supported by an SERC Advanced Fellowship.

Also special thanks to Jeremy Crowe in Edinburgh, who provided many useful insights about the training corpora, and the code for removing reported speech.

The system has been developed entirely within the POPLOG programming environment, combining Pop11, Prolog, Lisp and C. Our thanks to the POPLOG development and support team at Sussex and ISL for providing such a productive environment to work with.

References

[1] D. Allport. The TIC: parsing interesting text. In *Proceedings of the Second ACL Conference on Applied Natural Language Processing*, pages 211–218, 1988.

[2] E.J. Briscoe, C. Grover, B.K Boguraev, and J. Carroll. A formalism and environment for the development of a large grammar of English. *IJCAI-87*, 2:703–708, 1987.

[3] L. J. Cahill. Some reflections on the conversion of the TIC lexicon to DATR. In *Default Inheritance Within Unification-Based Approaches to the Lexicon*. Cambridge University Press, Cambridge, 1992.

[4] L.J. Cahill and R. Evans. An application of DATR: the TIC lexicon. In *Proceedings of the 9th European Conference on Artificial Intelligence*, pages 120–125, Stockholm, 1990.

[5] L.J. Cahill, R. Gaizauskas, and R. Evans. POETIC: a fully-implemented NL system for understanding traffic reports. In *Fully Implemented Natural Language Understanding Systems: Proceedings of the Trento Workshop, March 30, 1992 (IWBS Report No. 236)*, pages 86–99, IBM Institute for Knowledge Based Systems, Heidelberg, 1992.

[6] L. Carlson and S. Nirenburg. Practical world modeling for NLP applications. In *Proceedings of the Third Conference on Applied Natural Language Processing*, pages 235–236. Association for Computational Linguistics, 1992.

[7] E. Charniak and D. McDermott. *Introduction to Artificial Intelligence*. Addison-Wesley, Reading, Mass., 1985.

[8] K. Dahlgren. *Naive Semantics for Natural Language Understanding*. Kluwer, Boston, 1988.

[9] R. Evans and G. Gazdar. Inference in DATR. *Proceedings of the Fourth Conference of the European Chapter of the Association for Computational Linguistics*, 1989.

[10] R. Evans and G. Gazdar. The semantics of DATR. In A. Cohn, editor, *Proceedings of the Seventh Conference of the Society for the Study of Artificial Intelligence and Simulation of Behaviour*, pages 79–87. Pitman, London, 1989.

[11] R. Evans and A.F. Hartley. The traffic information collator. *Expert Systems: The International Journal of Knowledge Engineering*, 7(4):209–214, 1990.

[12] J.R. Hobbs, M. Stickel, P. Martin, and D. Edwards. Interpretation as abduction. In *Proceedings of the 26th Conference of the Association for Computational Linguistics*, Buffalo, N.Y., 1988.

[13] C. Mellish, D. Allport, A.F. Hartley, R. Evans, L. J. Cahill, R. Gaizauskas, and J. Walker. The TIC message analyser. Cognitive Science Research Paper 225, University of Sussex, 1992. Submitted for publication.

[14] C. S. Mellish, R. Evans, and J. Walker. The TIC project final report. Cognitive Science Research Paper 208, Cognitive and Computing Sciences, University of Sussex, 1991.

[15] R.C. Schank and R.P. Abelson. *Scripts, Plans, Goals and Understanding*. Lawrence Erlbaum, Hillsdale, N.J., 1977.

APPENDIX: DETAILED SYSTEM WALKTHROUGH – PAGE 1

PREPROCESSOR OUTPUT

1. DATE '241189'

2. SOURCE 'Jiji Press Ltd.'

3. NEWCONTEXT 1

4. BRIDGESTONE SPORTS CO . SAID FRIDAY IT HAS SET UP A JOINT VENTURE IN TAIWAN WITH A LOCAL CONCERN AND A JAPANESE TRADING HOUSE TO PRODUCE GOLF CLUBS TO BE SHIPPED TO JAPAN

5. THE JOINT VENTURE , BRIDGESTONE SPORTS TAIWAN CO . , CAPITALIZED AT 20 MILLION NEW TAIWAN DOLLARS , WILL START PRODUCTION IN JANUARY 1990 WITH PRODUCTION OF 20000 IRON AND \" METAL WOOD \" CLUBS A MONTH

6. THE MONTHLY OUTPUT WILL BE LATER RAISED TO 50000 UNITS , BRIDGESTON SPORTS OFFICIALS SAID

7. THE NEW COMPANY , BASED IN KAOHSIUNG , SOUTHERN TAIWAN , IS OWNED 75 PCT BY BRIDGESTONE SPORTS , 15 PCT BY UNION PRECISION CASTING CO . OF TAIWAN AND THE REMAINDER BY TAGA CO . , A COMPANY ACTIVE IN TRADING WITH TAIWAN , THE OFFICIALS SAID

8. BRIDGESTONE SPORTS HAS SO FAR BEEN ENTRUSTING PRODUCTION OF GOLF CLUB PARTS WITH UNION PRECISION CASTING AND OTHER TAIWAN COMPANIES

9. WITH THE ESTABLISHMENT OF THE TAIWAN UNIT , THE JAPANESE SPORTS GOODS MAKER PLANS TO INCREASE PRODUCTION OF LUXURY CLUBS IN JAPAN

10. CASE ALLUPPER

PARSER OUTPUT

```
[template_doc_date(z1, '241189')]

[template_doc_source(z1, 'Jiji Press Ltd.')]

[template_new_context(z1, 1)]

[company(e1), dbnlnfo(e1,[[value,'BRIDGESTONE CORPORATION | JAPAN '],
                          [orig_text,'BRIDGESTONE']])]
[company(e2), numberof(e2, *, 1)]
[tie_up(e3), numberof(e4, *, 1), location(e4), at(e3, e4),
 dbnlnfo(e4,[[value,'Taiwan (COUNTRY)'],[type,'COUNTRY'],
             [orig_text,'TAIWAN']])]
[ecentity(e5), numberof(e5, *, 1), other_ecentity(e6),
 numberof(e6, *, 1), nationality(e6, japan)]
[location(e7), dbnlnfo(e7,[[value,'Japan (COUNTRY)'],
                           [type,'COUNTRY'],[orig_text,'JAPAN']])]

(tie_up(e1), numberof(e1, *, 1)]
[company(e2), dbnlnfo(e2,[[value,'BRIDGESTONE CORPORATION | JAPAN '],
                          [orig_text,'BRIDGESTONE']])]
[company(e3), dbnlnfo(e3,[[value,'TAIWAN | 886SOUTH'],
                          [orig_text,'TAIWAN CO']])]
[location(e4), dbnlnfo(e4,[[value,'Taiwan (COUNTRY)'],
                           [type,'COUNTRY'],[orig_text,'TAIWAN']])]
[monetary_amount_number(e5, e6), monetary_amount(e5),
 monetary_amount_unit(e5, dollars)]
[month(e7, 1), time(e7), year(e7, 1990)]

[ecentity(e1), numberof(e1, *, 50000)]
[person(e2)]

[company(e1), numberof(e1, *, 1)]
[location(e2), dbnlnfo(e2,[[value,'Taiwan (COUNTRY)'],
                           [type,'COUNTRY'],[orig_text,'TAIWAN']])]
[company(e3), dbnlnfo(e3,[[value,'BRIDGESTONE CORPORATION | JAPAN '],
                          [orig_text,'BRIDGESTONE']])]
[location(e4), dbnlnfo(e4,[[value,'Taiwan (COUNTRY)'],
                           [type,'COUNTRY'],[orig_text,'TAIWAN']])]
[company(e5),dbnlnfo(e5,[[value,'TAGA | 886SOUTH'],[orig_text,'TAGA CO']])]
[company(e6), dbnlnfo(e6,[[value,'A | 886SOUTH'],[orig_text,'A COMPANY']])]
[location(e7), dbnlnfo(e7,[[value,'Taiwan (COUNTRY)'],
                           [type,'COUNTRY'],[orig_text,'TAIWAN']])]
[person(e8)]

[company(e1), dbnlnfo(e1,[[value,'BRIDGESTONE CORPORATION | JAPAN '],
                          [orig_text,'BRIDGESTONE']])]
[company(e2), dbnlnfo(e2,[[value,'CASTING | 886SOUTH'],
                          [orig_text,CASTING']])]
[location(e3), dbnlnfo(e3,[[value,'Taiwan (COUNTRY)'],
                           [type,'COUNTRY'],[orig_text,'TAIWAN']])]
[company(e4), numberof(e4, *, 1)]

[location(e1), dbnlnfo(e1,[[value,'Taiwan (COUNTRY)'],
                           [type,'COUNTRY'],[orig_text,'TAIWAN']])]
[ecentity(e2), numberof(e2, *, 1)]
[company(e3), numberof(e3, *, 1)]
[location(e4), dbnlnfo(e4,[[value,'Japan (COUNTRY)'],
                           [type,'COUNTRY'],[orig_text,'JAPAN']])]

[template_doc_case(z1, allupper)]
```

DISCOURSE INTERPRETER OUTPUT

```
Model1 :=
 z1 <-- template(_)
    Props: [template_doc_date(z1, 241189), template_doc_ar(z1, 659.)]

Model2 :=
 z1 <-- template(_)
    Props: Model1:Props + [template_doc_source(z1, 'Jiji Press Ltd.')]

Model3 :=
 z1 <-- template(_)
    Props: Model2:Props + [template_new_context(z1, 1)]

Model4 := Model3 +
 z2 <-- company(_)
    Props: [numberof(z2, *, 1), distinct(z2, z5),ecentity_type(z2, COMPANY),
            dbnlnfo(z2,[[value,'BRIDGESTONE CORPORATION | JAPAN'],
                [orig_text,'BRIDGESTONE']]),
            ecentity_core_name(z2,'BRIDGESTONE CORPORATION | JAPAN'),
            tie_up_ecentity(z3, z2), ecentity_name(z2, 'BRIDGESTONE')]
 z3 <-- tie_up(_)
    Props: [tie_up_ecentity(z3,z5),tie_up_activity(z3,z7),
            numberof(z3,*,1), tie_up_ecentity(z3,z2),at(z3,z6)]
 z6 <-- location(_)
    Props: [dbnlnfo(z6,[[value,'Taiwan (COUNTRY)'],
                [type,COUNTRY],[orig_text,'TAIWAN']]), at(z3, z6)]
 z5 <-- other_ecentity(_)
    Props: [numberof(z5, *, 1), nationality(z5, japan),distinct(z2, z5),
            tie_up_ecentity(z3,z5),ecentity_nationality(z5,'Japan (COUNTRY)')]

Model5 := Model4 + (z1,z2,z3) +
 z1 <-- template(_)
    Props: Model4:Props + [template_content(z1, z3)]
 z2 <-- company(_)
    Props: Model4:Props + [distinct(z2, z113)]
 z3 <-- tie_up(_)
    Props: Model4:Props + [template_content'z1,z3),tie_up_ecentity(z3,z113]
 z11 <-- company(_)
    Props: [dbnlnfo(z11,[[value,'TAIWAN | 886SOUTH'],
                [orig_text,'TAIWAN CO']]), ecentity_name(z11,'TAIWAN CO'),
            ecentity_core_name(z11,'TAIWAN | 886SOUTH'),distinct(z2, z11),
            ecentity_type(z11, COMPANY), tie_up_ecentity(z3, z11)]

Model6 := Model5 +
 z14 <-- person(_)
    Props: [ecentity_type(z14, PERSON), numberof(z14, *, 50000)]

Model7 := Model6 +
 z15 <-- company(_)
    Props: [ecentity_type(z15, COMPANY), ecentity_name(z15,'TAGA CO'),
            ecentity_core_name(z15,'TAGA | 886SOUTH'),
            dbnlnfo(z15,[[value,'TAGA | 886SOUTH'],[orig_text,'TAGA CO']])]
 z16 <-- company(_)
    Props: [ecentity_type(z16, COMPANY), ecentity_name(z16,'A CO'),
            ecentity_core_name(z16,'A | 886SOUTH'),
            dbnlnfo(z16,[[value,'A | 886SOUTH'], [orig_text,'A COMPANY']])]

Model8 := Model7 - (z2) +
 z2 <-- company(_)
    Props: Model7:Props + [numberof(z2, >, 1)]
 z17 <-- company(_)
    Props: [ecentity_type(z17, COMPANY), ecentity_name(z17,'CASTING'),
            ecentity_core_name(z17,'CASTING | 886SOUTH'),
            dbnlnfo(z17,[[value,'CASTING | 886SOUTH'],[orig_text,'CASTING']])]

Model9 := Model8

Model10 := Model9 - (1) +
z1 <-- template(_)
    Props: Model9:Props + [template_doc_case(z1, allupper)]
```

APPENDIX: DETAILED SYSTEM WALKTHROUGH – PAGE 2

REMARKS

'z1 <-- template(_):' z1 is an instance of the type template with properties as specified.

Fixed field information in sentences 1,2,3 and 10 goes straight through to the model -- the parser outputs are fixed.

'newcontext' provides a mechanism for syntactic recognition of multiple distinct JV reports per document.

BRIDGESTONE and CO are parsed as separate companies but get resolved back together. LOCAL COMPANY is resolved in too. So we get a JV between BRIDGESTONE and JAPANESE TRADING HOUSE and presuppose some activity. But GOLF CLUBS is unknown.

A piecemeal parse, but the new JV gets merged with the existing one, as TAIWAN CO becomes a partner. With mixed case, many of these company names would be much more accurate.

Very little happens here - we don't have a spelling corrector to fix up the name, and don't find much else of interest

This sentence gives a variety of companies, but little structure. The net result is just the addition of two companies not connected to anything else

Note here that CASTING is recognised, because we have seen it before.

The last two add nothing of interest. Finally we complete the model adding some default information about relationships and industry. From this completed model we build the template.

COMPLETED DISCOURSE MODEL

```
z1 <-- template(_17)
    Props: [template_doc_date(z1,261189),template_doc_nr(z1,0592),
            template_doc_source(z1,'Jiji Press Ltd.'),
            template_content(z1,z3),template_doc_case(z1,slipper),
            template_new_content(z1,17)]
z2 <-- company(_15)
    Props: [numberof(z2, >, 1),ecentity_type'z2, COMPANY),
            ecentity_name(z2,'BRIDGESTONE'), distinct(z2, z11),
            tie_up_ecentity(z3, z2), distinct(z2, z5),
            ecentity_core_name(z2,'BRIDGESTONE CORPORATION | JAPAN'),
            dbminfo(z2,[[value,'BRIDGESTONE CORPORATION | JAPAN'],
                [orig_text,'BRIDGESTONE']]), numberof(z2, =, 1),
            ecentity_ecentity_relationship(z2, z10),
            ecentity_relationship_ecentity1'z10, z2)]
z3 <-- tie_up(_2)
    Props: [tie_up_status(z3, existing), at(z3, z4),
            tie_up_ecentity(z3, z5), numberof(z3, =, 1),
            template_content(z1, z3), tie_up_activity(z3, z7),
            tie_up_ecentity(z3, z11), tie_up_ecentity(z3, z2)]
z4 <-- location(_14)
    Props: [dbminfo(z4,[[value,'Taiwan (COUNTRY)'],
            [type,COUNTRY],[orig_text,'TAIWAN']]),at(z3,z4)]
z5 <-- other_ecentity(_1)
    Props: [ecentity_type(z5, COMPANY),
            ecentity_ecentity_relationship(z5, z10),
            ecentity_relationship_ecentity1(z10, z5),
            numberof(z5, =, 1), distinct(z2,z5),
            nationality(z5, japan), tie_up_ecentity(z3, z5),
            ecentity_nationality(z5,'japan (COUNTRY)')]
z6 <-- location(_16)
    Props: [dbminfo(z6,[[value,'Japan (COUNTRY)'],
                [type,COUNTRY],[orig_text,'JAPAN']])]
z7 <-- activity(_6)
    Props: [activity_activity_site(z7, z8),
            activity_industry'z7, z9), tie_up_activity(z3, z7)]
z8 <-- industry(_7)
    Props: [industry_industry_type(z8, production),
            industry_product_service(z8,z10),
            activity_industry(z7,z8)]
z9 <-- activity_site(_8)
    Props: [activity_activity_site(z7, z9)]
z10 <-- product_service(_9)
    Props: [industry_product_service(z8, z10)]
z11 <-- company(_3)
    Props: [ecentity_type(z11, COMPANY),
            ecentity_ecentity_relationship'z11, z10),
            ecentity_relationship_ecentity1'z10, z11),
            ecentity_core_name(z11,'TAIWAN | 99KEOWB'),
            dbminfo(z11, [[value,'TAIWAN | 99KEOWB'],
                [orig_text,'TAIWAN CO']]),tie_up_ecentity(z3,z11),
            distinct(z2, z11), ecentity_name(z11,'TAIWAN CO')]
z12 <-- monetary_amount(_4)
    Props: [monetary_amount_number(z12, z22),
            monetary_amount_unit'z12, dollars)]
z13 <-- time(_5)
    Props: [month(z13, 1), year(z13, 1990)]
z14 <-- person(_12)
    Props: [ecentity_type(z14, PERSON), numberof(z14, =, 50000)]
z15 <-- company(_10)
    Props: [ecentity_type(z15,COMPANY),ecentity_name(z15,'TAGA CO'),
            ecentity_core_name(z15,'TAGA | 99KEOWB'),
            dbminfo(z15,[[value,'TAGA | 99KEOWB'],
                [orig_text,'TAGA CO']])]
z16 <-- company(_11)
    Props: [ecentity_type(z16,COMPANY),ecentity_name(z16,'A CO'),
            ecentity_core_name(z16,'A | 99KEOWB'),
            dbminfo(z16,[[value,'A | 99KEOWB'],
                [orig_text,'A COMPANY']])]
z17 <-- company(_13)
    Props: [dbminfo(z17, [[value,'CASTING | 99KEOWB'],
                [orig_text, CASTING]]),
            ecentity_core_name(z17,'CASTING | 99KEOWB'),
            ecentity_type(z17, COMPANY),
            ecentity_name(z17,'CASTING')]
z18 <-- ecentity_relationship(_18)
    Props: [ecentity_relationship_status(z18, current),
            ecentity_relationship_role2of(z18, partner),
            ecentity_ecentity_relationship(z2, z18),
            ecentity_ecentity_relationship(z11, z18),
            ecentity_ecentity_relationship(z5, z18),
            ecentity_relationship_ecentity1(z18, z2),
            ecentity_relationship_ecentity1(z18, z11),
            ecentity_relationship_ecentity1(z18, z5)]
```

RESPONSE TEMPLATE

```
<TEMPLATE-0592-1> :=
    DOC NR: 0592
    DOC DATE: 261189
    DOCUMENT SOURCE: "Jiji Press Ltd."
    CONTENT: <TIE_UP_RELATIONSHIP-0592-1>
<TIE_UP_RELATIONSHIP-0592-1> :=
    TIE-UP STATUS: EXISTING
    ENTITY: <ENTITY-0592-1>
            <ENTITY-0592-3>
            <ENTITY-0592-2>
    ACTIVITY: <ACTIVITY-0592-1>
<ENTITY-0592-1> :=
    NATIONALITY: Japan (COUNTRY)
    TYPE: COMPANY
    ENTITY RELATIONSHIP: <ENTITY RELATIONSHIP-0592-1>
<ENTITY-0592-2> :=
    NAME: BRIDGESTONE
    TYPE: COMPANY
    ENTITY RELATIONSHIP: <ENTITY RELATIONSHIP-0592-1>
<ENTITY-0592-3> :=
    NAME: TAIWAN CO
    TYPE: COMPANY
    ENTITY RELATIONSHIP: <ENTITY RELATIONSHIP-0592-1>
<INDUSTRY-0592-1> :=
    INDUSTRY-TYPE: PRODUCTION
<ENTITY_RELATIONSHIP-0592-1> :=
    ENTITY1: <ENTITY-0592-2>
             <ENTITY-0592-3>
             <ENTITY-0592-1>
    REL OF ENTITY2 TO ENTITY1: PARTNER
    STATUS: CURRENT
<ACTIVITY-0592-1> :=
    INDUSTRY: <INDUSTRY-0592-1>
```

KEY TEMPLATE

```
<TEMPLATE-0592-1> :=
    DOC NR: 0592
    DOC DATE: 261189
    DOCUMENT SOURCE: "Jiji Press Ltd."
    CONTENT: <TIE_UP_RELATIONSHIP-0592-1>
    DATE TEMPLATE COMPLETED: 251192
    EXTRACTION TIME: 0
    COMMENT: / "TOOL_VERSION: NUME.7.1.def.def"
             / "FILLRULES_VERSION: EJV.7.1"
<TIE_UP_RELATIONSHIP-0592-1> :=
    TIE-UP STATUS: EXISTING
    ENTITY: <ENTITY-0592-1>
            <ENTITY-0592-2>
            <ENTITY-0592-3>
    JOINT VENTURE CO: <ENTITY-0592-4>
    OWNERSHIP: <OWNERSHIP-0592-1>
    ACTIVITY: <ACTIVITY-0592-1>
<ENTITY-0592-1> :=
    NAME: BRIDGESTONE SPORTS CO
    ALIASES: "BRIDGESTONE SPORTS"
             "BRIDGESTON SPORTS"
    NATIONALITY: Japan (COUNTRY)
    TYPE: COMPANY
    ENTITY RELATIONSHIP: <ENTITY_RELATIONSHIP-0592-1>
<ENTITY-0592-2> :=
    NAME: BRION PRECISION CASTING CO
    ALIASES: "BRION PRECISION CASTING"
    LOCATION: Taiwan (COUNTRY)
    NATIONALITY: Taiwan (COUNTRY)
    TYPE: COMPANY
    ENTITY RELATIONSHIP: <ENTITY_RELATIONSHIP-0592-1>
<ENTITY-0592-3> :=
    NAME: TAGA CO
    NATIONALITY: Japan (COUNTRY)
    TYPE: COMPANY
    ENTITY RELATIONSHIP: <ENTITY_RELATIONSHIP-0592-1>
<ENTITY-0592-4> :=
    NAME: BRIDGESTONE SPORTS  TAIWAN  CO
    LOCATION: "HAOHSIUNG" (99KEOWB) Taiwan (COUNTRY)
    TYPE: COMPANY
    ENTITY RELATIONSHIP: <ENTITY_RELATIONSHIP-0592-1>
<INDUSTRY-0592-1> :=
    INDUSTRY-TYPE: PRODUCTION
    PRODUCT/SERVICE: ("30 "20,000 IRON AND "METAL WOOD" [CLUBS]"
             / ("30 "GOLF [CLUBS]")
             / ("30 "GOLF [CLUBS] TO BE SHIPPED TO  JAPAN")
<ENTITY_RELATIONSHIP-0592-1> :=
    ENTITY1: <ENTITY-0592-1>
             <ENTITY-0592-2>
             <ENTITY-0592-3>
    ENTITY2: <ENTITY-0592-4>
    REL OF ENTITY2 TO ENTITY1: CHILD
    STATUS: CURRENT
<ACTIVITY-0592-1> :=
    INDUSTRY: <INDUSTRY-0592-1>
    ACTIVITY-SITE: (Taiwan (COUNTRY) <ENTITY-0592-4>)
    START TIME: <TIME-0592-1>
<TIME-0592-1> :=
    DURING: 0190
<OWNERSHIP-0592-1> :=
    OWNED: <ENTITY-0592-4>
    TOTAL-CAPITALIZATION: 20000000 TWD
    OWNERSHIP-%: (<ENTITY-0592-3> 10)
             (<ENTITY-0592-2> 15)
             (<ENTITY-0592-1> 75)
```