CRL/BRANDEIS: DESCRIPTION OF THE *DIDEROT* SYSTEM AS USED FOR MUC-5

Jim Cowie, Louise Guthrie, Wang Jin, Rong Wang, Takahiro Wakao
Computing Research Laboratory, New Mexico State University
Email: jcowie@nmsu.edu &
James Pustejovsky, Scott Waterman
Computer Science Department, Brandeis University
Email: jamesp@cs.brandeis.edu

Abstract

This report describes the major developments over the last six months in completing the Diderot information extraction system for the MUC-5 evaluation.

Diderot is an information extraction system built at CRL and Brandeis University over the past two years. It was produced as part of our efforts in the Tipster project. The same overall system architecture has been used for English and Japanese and for the micro-electronics and joint venture domains.

The past history of the system is discussed and the operation of its major components described. A summary of scores at the 24 month workshop is given and the performance of the system on the texts selected for the system walkthrough is discussed.

INTRODUCTION

The Computing Research Laboratory at New Mexico State University, in collaboration with Brandeis University, was one of four sites selected to develop systems to extract relevant information automatically from English and Japanese texts. The systems produced by the Tipster research groups have already been evaluated at 12 and 18 months into the project. The performance of the Diderot System has improved both for English and Japanese. The performance in Japanese, however, is still far ahead of our English performance.

The Tipster project is, without a doubt, the largest scale Applied Natural Language Processing task yet undertaken anywhere in the world. The government data preparation effort involved the selection and analysis of more than 5,000 individual texts. The results of this analysis have been used to develop and test the systems produced by each site. The software used to support this human extraction task, both for English and Japanese, was developed and supported by the CRL under a separate subcontract.

Because of the emphasis on different languages and different subject areas the research has focused on the development of general purpose, re-usable techniques. The CRL/Brandeis group have implemented statistical methods for focusing on the relevant parts of texts, programs which recognize and mark names of people, places and organizations and also dates. The actual analysis of the critical parts of the texts is carried out by a parser controlled by lexical structures for the 'key' words in the text. To extend the system's coverage of English and Japanese some of the content of these lexical structures was derived from machine readable dictionaries. These were then enhanced with information extracted from corpora.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE
1993

2. REPORT TYPE

3. DATES COVERED
00-00-1993 to 00-00-1993

4. TITLE AND SUBTITLE
CRL/Brandeis: Description of the DIDEROT System as Used for MUC-5

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Department of Computer Science,New Mexico State University,PO Box 30001,Las Cruces,NM,88003-8001

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR’S ACRONYM(S)

11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:

a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
unclassified	**unclassified**	**unclassified**		**19**	

The system has already been evaluated in the 4th Message Understanding Conference (MUC-4) where it was required to extract information from 200 texts on South American terrorism. Considering the very short development time allowed for this additional domain the system performed adequately. The system was then adapted to handle the business domain and also to process Japanese texts. Further extensions to the system allowed it to process texts on micro-electronics development. Performance at the 12 and 18 month evaluations was good for Japanese, but less good for English where we have been attempting to automate much of the development process. A more pragmatic approach was adopted for the final 24 month evaluation, using the same hand-crafted techniques for English as had been used for Japanese.

We estimate the amount of effort used directly to build the systems described here is around sixty man months.

MAIN RESEARCH OBJECTIVES

Our objectives in this research have been as follows:

- to develop and implement a language-independent framework for lexical semantic representation, and develop and implement a robust integration of that framework into a language-independent theory of semantic processing;
- to investigate and implement language independent techniques for automating the building of lexical knowledge bases from machine readable resources;
- to implement statistical tools for the tuning of lexical structures to specific domains;
- to implement the use of language independent statistical techniques for identifying relevant passages of documents for more detailed analysis;
- to develop and implement a set of robust multi-pass finite-state feature taggers;
- to develop and implement the equivalent methods for Japanese.

SYSTEM OVERVIEW

An outline of the functions of the main system modules are given here. This is intended to provide a context for the more detailed description of each module which follows. The structures of the Japanese and English systems are very similar. In the examples of intermediate output either Japanese or English may be shown. The system architecture is shown in figure 1.

The input text to the system is processed by three independent pre-processing modules:

- A chain of finite-state feature taggers - these mark: names, organization names, place names, date expressions and other proper names (depending on the domain),
- A part of speech tagger,
- A statistically based determiner of text relevance (micro only).

If the statistical determination rejects the text processing proceeds to the final output stage and an empty template is produced. Otherwise the results of the other two stages are converted to Prolog facts and these then pass into the head of a chain of processes each of which gives rise to further refinements of the text:

- Merge - Here semantic tags, which may mark phrasal units, are merged with POS tags, which mark individual words.
- Compound noun recognizer - this groups words and phrases into compound nouns using POS and semantic information.

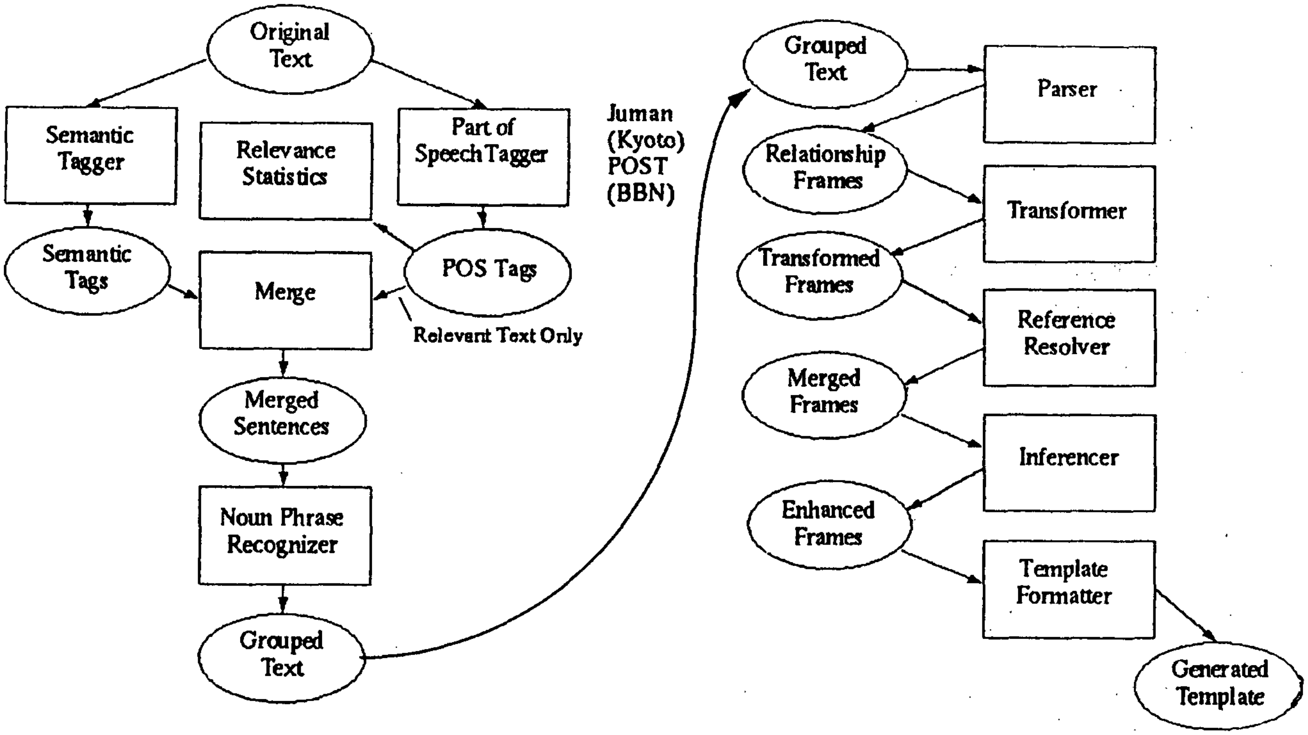


Figure 1: System Overview

- Parser - the relevant paragraph information is used to select which sentences to process further. The sentences containing the marked up noun-phrase groups are then parsed to produce a partially completed representation of the relevant semantic content of the sentence (frames).
- Reference resolver - the frames are then merged based on name matching and noun compounds beginning with definite articles.
- Template formatter - this transforms the resolved frames into the final output form.

Statistical Filtering Techniques

Statistical information is used to predict whether a text holds important information that is relevant to completing a template. This allows the parser to skip non-relevant texts. This is based on word lists which are derived from training on relevant and irrelevant texts. The theoretical results on which the method [6] is based assure us that documents can be classified correctly if appropriate sets of words can be chosen for each document type. The method was only applied to the micro domain for MUC-5 as almost all texts in the joint venture domain are relevant and the use of this statistical method is essentially a way of improving precision in text filtering.

The results for the micro electronics domain for text filtering are 84% recall and 90% precision (73 and 83 at 18 month) for Japanese, and 78% recall and 83% precision (77 and 76 at 18 month) for

English.

Semantic Tagging

This component is based on a pipeline of programs. These are all written in 'C' or flex. It marks organization names, human names, place names, date expressions, equipment names, process types and a variety of measurements (including money). Many of these have converted forms and additional values attached by the tagger.

The tagging programs use three separate methods —

- Direct recognition of already known unambiguous names, using a longest string match.
- Recognition using textual patterns only.
- Two pass method marking ambiguous, but potential names, and subsequently verifying they fit a pattern.
- final pass recognizing short forms and isolated occurrences of names not in a strong context

The system uses the case of letters used when available. The final text is tagged using SGML-like markers.

```
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.
```

```
<organ> BRIDGESTONE SPORTS CO. {type([{}[entity_type,'COMPANY']{}])} <\endorgan> said
<date> FRIDAY{type([{}[date,'241189']{}])} <\enddate> it has set up a joint
venture in <country> TAIWAN {type([{}[nationality,'TAIWAN']{}])}
<\endcountry> with a local concern and a
<country>japanese {type([{}[nationality,'JAPAN'], [word_type,sp_noun]{}])} <\endcountry>
trading house to produce golf clubs to be shipped
to <country> JAPAN {type([{}[nationality,'JAPAN']{}])} <\endcountry>.
```

At this point the tags are converted into Prolog facts:

```
organ('BRIDGESTONE SPORTS CO.',type([[entity_type, 'COMPANY']])),
res('said',type([[undefined,'said']])),
time('FRIDAY',type([[date_adverb,'UNSPEC'],[date,'241189']])),
cs('it',type([[it,[pron]]])),
cs('has',type([[has,[pastv,presv]]])),
gls('set up',type([['set up',v]])),
cs('a',type([[a,[determiner]]])),
gls('joint venture',type([['joint venture',comp]])),
date_adverb('in',type([[date_adverb,during]])),
country('TAIWAN',type([[nationality,'TAIWAN']])),
cs('with',type([[with,[prep]]])),
```

The Japanese system preprocesses the article to change the original encoding (Shift JIS) to EUC for a given article. The original and unsegmented text goes through a series of taggers for known names, i.e. organizations, places, GLS verbs. This process is exactly the same as in the English system. The next step is to tag organization, personal and place names which are not known to the system. These are detected by using local context, using Japanese-specific patterns, which use particles, specific words and the text tags to recognize the unknown names. In addition, date expressions are tagged and changed into the normalized form. Date expressions in the Japanese articles seem straightforward, for example, '20 nichi' (20 day) is used even if the document date is 21st and 20th can be expressed as 'yesterday', and this convention 'XX day' (where XX is a number) to express a date is consistently used in the articles. Era names such as '昭和' (Showa) or '平成' (Heisei) are Japanese specific and the year in the era, e.g. '' (Showa 60th year), is correctly recognized and normalized. Here is the first sentence of a typical article after the tagging process.

```
<\organ> 東京海上火災保険 {type([[entity_type,'COMPANY']])}
<\endorgan> は <\time>今月から {type([[date_adverb,after],
 [date,'8501']])}
<\endtime>  <\organ> 大和証券 {type([[entity_type,'COMPANY']])}
<\endorgan> と  <\gls> 提携して {type(['提携する',v])} <\endgls>
```

Just as for the English system this is then converted into the form of Prolog facts ready to be read into the merging phase.

Part-Of-Speech Tagging

English text is also fed through the POST part of speech tagger. This attaches the Penn Treebank parts of speech to the text. The output is converted to Prolog facts. The Japanese text is segmented with part-of-speech information by the JUMAN program, which was developed by Kyoto University. The following is the result for exactly the same sentence. The segmented units are converted to Prolog facts ready to input to the next stage.

```
juman('東京','proper_noun').
juman('海上','proper_noun').
juman('火災','normal_noun').
juman('保険','normal_noun').
juman('は','topic_particle').
juman('今月','normal_noun').
juman('から','case_particle').
juman('大和','normal_noun').
juman('証券','normal_noun').
juman('と','case_particle').
juman('提携','noun_verb').
juman('して','verb').
```

Merging

The semantic and syntactic information are merged to give lexical items in the form of triples. The merging is done in such a way that if it is not possible to match up words (eg due to different treatments of hyphens) a syntactic tag of 'UNK' is allocated and merging continues with the next word.

Noun Phrase Grouping

Noun phrases are identified by scanning back through a sentence to identify head nouns. Both semantically and syntactically marked units qualify as nouns. The grouping stops when closed class words are encountered. A second forward pass gathers any trailing adjectives. The main use of the noun phrase in the present system is to attach related strings to company names to help with the reference resolution. They are also used by a retrieval process which uses the string to determine the SIC code industry type.

A similar grouping is carried out for Japanese.

```
noun_phrase([[undefined,house]],
    [unit(cs,a,type([[a,[determiner]]]),['DT']),
     unit(country,japanese,type([[nationality,'JAPAN'],[word_type,sp_noun]]),['JJ']),
     unit(res,trading,type([[undefined,trading]]),['NN']),
     unit(res,house,type([[undefined,house]]),['NN'])])
```

```
noun_phrase(money,
    [unit(num,'20',type([[num_value,20]]),['CD']),
     unit(num,million,type([[num_value,1000000]]),['CD']),
     unit(money,'NEW TAIWAN dollars',type([[denom,'TWD']]),['NP','NP','NNS'])])
```

Parsing

The parser has GLS cospecification patterns built into it. It uses these and ancillary rules for the recognition of semantic objects to fill a frame format which was given as an application specific field in the GLS entry. The frame formats provide a bridge between the sentence level parse and the final template output. Semantic objects are named in the cospecification and special rules which handle type checking, conjunction and co-ordination are used to return a structure for the object. The following shows an example of a tie-up between two companies. The child company is unmatched, shown by an underscore. The parser has grouped a date with one of the companies. The tie-up status is provided by the GLS template semantics.

```
prim_tie_up(1,1,[
 [[f(name,_9947,[unit(organ,'東京海上火災保険',
     type([[entity_type,'COMPANY']]),[proper_noun])]),
   f(entity_type,_9953,[unit(organ,'東京海上火災保険',
     type([[entity_type,'COMPANY']]),[proper_noun])])]],
 [[f(name,_10102,[unit(organ,'大和証券',
    type([[entity_type,'COMPANY']]),[proper_noun])]),
   f(entity_type,_10108,[unit(organ,'大和証券',
    type([[entity_type,'COMPANY']]),[proper_noun])]),
   f(time,_10114,[unit(time,'今月から',
    type([[date_adverb,after],[date,'8501']]),[proper_noun])])]],_,
  [f(tie_up_status,existing,[])]).
```

Reference Resolution

The task of this component is to gather all the relevant information scattered in a text together. The major task is to resolve reference or anaphora. For the current application only references between tie-up events, between entities, and between entity relations are considered.

Since entities are expressed in noun phrases, the references for entities are resolved by resolving the reference between noun phrases. Since the entity can either be referred to by definite or indefinite noun phrase or by name, it is necessary to detect the reference between two definite or indefinite noun phrases, between two names, as well as between one name and one definite or indefinite noun phrase. All entities are represented as frames of the form:

```
entity(Sen#, Para#, Noun-phrase, Name,
       Location, Nationality,
       Ent-type, alias-list, np-list).
```

The reference between two entities is resolved by looking at the similarity between their names and/or their noun phrases. Since companies are often referred by their nationality or location, the Location and Nationality slot fillers in the entity frame also contribute to the reference resolution. Some special noun phrases which refer to some particular role of a tie-up (*the newly formed venture*in particular) are also recognized and resolved. For example, a phrase which refers to the child entity, such as *'the new company'* or *'the venture'*, will be recognized and merged with the child of the tie-up event in focus. A stack of entities found in the text is maintained.

Definite noun phrases can only be used for local reference. So they can only be used to refer to entities involved in the tie-up event which is in focus. On the contrary, names can be used for both local and global reference, so they can refer to any entity referred to before in the text.

When a reference relation between two entities is resolved they are merged to create one single entity which contains all the information about that particular entity.

Since a tie-up is generally referenced by an entire sentence rather than a single noun phrase, the reference of tie-up events is handled by resolving the reference between its participants and some other information mentioned about the event. Other heuristics are also applied. These mostly block the overapplication of merging. For example, two tie-ups cannot be merged if their dates are different; similarly, entities with different locations will not be merged. There are currently two types of text structures which are considered. In the first type, one tie-up-event is in focus until the next one is mentioned and after the new one is mentioned the old one will not be mentioned again. In the second type, a list of tie-up-events are mentioned shortly in one paragraph, and more details of each event are given sequentially later. Finally, when the reference between two tie-ups is resolved they will also be merged to form a single tie-up event. The final result is a set of new frames which are linked in such a way as to reduce the requirement on the final stage of maintaining pointers to the various objects.

With the exception of the use of definite articles —an obvious cross-linguistic difference between the languages studied— the reference resolution process for Japanese is identical to English. The resolved entities, entity-relation, and tie-up for a typical text are shown below.

```
final_entity(2,[f(name,['大','和','証','券'],'UNSPEC'),
  f(entity_type,'COMPANY','UNSPEC'),
  f(industry_product,'(63 "財形年金")',wj),
  f(time,[after,'8501'],wj),f(entity_relationship,1,inf),
  f(entity_relationship,3,inf)]).
final_entity(9,[f(name,['東','京','海','上',火,災,'保',険],'UNSPEC'),
  f(entity_type,'COMPANY','UNSPEC'),f(name,['東','京','海','上'],
  'UNSPEC'),
  f(entity_relationship,1,inf),f(entity_relationship,3,inf)]).
final_rel(1,[9,2],'UNSPEC','PARTNER','UNSPEC').
final_tie_up(1,[9,2],'UNSPEC','UNSPEC','UNSPEC',existing,'UNSPEC',1,
  'UNSPEC').
```

The system uses character-based rules for identifying aliases. For example, if a company name starts with '日立' (Hitachi) as in '日立製 作所' (Hitachi Manufacturing), then the system looks for the string '日立' (Hitachi) or the first two characters of the company name as its alias.

Template Formatting

The final stage generates sequence numbers and incorporates document numbers into the labels. It also eliminates objects which are completely empty. The final output from the English system example text, #0592, is shown below.

```
<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-0592-3>
   JOINT VENTURE CO: <ENTITY-0592-1>
   OWNERSHIP: <OWNERSHIP-0592-1>
```

```
<ENTITY-0592-1> :=
   NAME: BRIDGESTONE SPORTS TAIWAN CO
   ALIASES: "BRIDGESTONE SPORTS"
   TYPE: COMPANY
   ENTITY RELATIONSHIP: <ENTITY_RELATIONSHIP-0592-1>
<ENTITY-0592-3> :=
   NAME: BRIDGESTONE SPORTS CO
   ALIASES: "BRIDGESTONE SPORTS"
   TYPE: COMPANY
   ENTITY RELATIONSHIP: <ENTITY_RELATIONSHIP-0592-1>
<ENTITY_RELATIONSHIP-0592-1> :=
   ENTITY1: <ENTITY-0592-3>
   ENTITY2: <ENTITY-0592-1>
   REL OF ENTITY2 TO ENTITY1: CHILD
   STATUS: CURRENT
<OWNERSHIP-0592-1> :=
   OWNED: <ENTITY-0592-1>
   TOTAL-CAPITALIZATION: 20000000 TWD
   OWNERSHIP-%: (<ENTITY-0592-3> 75 )
```

SUMMARY OF PERFORMANCE

Current performance for the CRL English Tipster systems as evaluated for the fifth Message Understanding Conference (MUC-5) are given in the appendix to this paper. All the scores have improved since the 18 month Tipster evaluation. The scores for Japanese, using an identical architecture, but with much more intensive human tuning, are much higher.

We feel the huge difference between performance in Japanese and English is principally due to one person being dedicated for Japanese to running and tuning the system. All other personnel were working on particular components to be used first in the English and then in the Japanese system and no one person was repeatedly testing the operation of the English System. Another difference might be due to the focus of effort on automatic and semi-automatic pattern generation for the English systems, a process which was not attempted for Japanese development. Although this differential would appear to speak slightly against these particular automatic development techniques, we feel that additional testing and refinement of the patterns would have brought the scores more in line with the Japanese systems, since they use the same architecture.

KNOWLEDGE ENGINEERING

The DIDEROT system has involved the development of a significant amount of diverse knowledge. This consisted of the automatic construction of the lexicon and partial syntactic forms for a given language and domain, along with subsequent tuning and refinement. Human tuning was also needed for both languages.

This off-line component included the derivation of vocabulary automatically from machine readable sources, and made use of statistically-based techniques to determine the relevant domain-dependent vocabulary of words and phrases from text samples.

Statistical techniques were used extensively to assist in the development of the various lexicons of the Tipster system. Initially, a simple frequency count of the tokens in the initial corpora was used to highlight those words which should be targeted, essentially determining what core vocabulary elements are of most importance. In general, the lexical structures used by the system can be thought of as providing for the shallowest possible semantic decomposition while still capturing significant generalizations about how words relate conceptually to one another.

Deriving the Lexicon from Machine-Readable Resources

The lexical knowledge base consists of lexical items called generative lexical structures (GLSs), after Pustejovsky[9]. This model of semantic knowledge associated with words is based on a system of generative devices which is able to recursively define new word senses for lexical items in the language. For this reason, the algorithm and associated dictionary is called a *generative lexicon.* The lexical structures contain conventional syntactic and morphological information along with detailed typing information about arguments

The creation of the GLS lexicon begins with the printer's tape of the Longman Dictionary of Contemporary English (LDOCE), Proctor[8]. This was parsed and analysed by the CRL lexical group to give a tractable formatted version of LDOCE called LEXBASE[5]. LEXBASE contains syntactic codes, inflectional variants, and boxcodes, selectional information for verbs and nouns, indicating generally what kind of arguments are well-formed with that lexical item. A GLS entry is automatically derived from LEXBASE by parsing the LEXBASE format for specific semantic information [11]. The most novel aspect of this conversion involves parsing the example sentences as well as parenthetical texts in the definition. This gives a much better indication of argument selection for an item than do the the boxcodes alone. For example, the verb *market* is converted into the following GLS entry as a result of this initial mapping.

```
gls(market,
   syn([type(v),
        code(gcode_t1),
        eventstr([]),
        ldoce_id(market_1_1),
        caseinfo([subcat1(A1),
                  subcat2(A2),
                  case(A1,np),
                  case(A2,np)]),
        inflection([ing(marketing),
                    pastp(marketed),
                    plpast(marketed),
                    singpast1(marketed),
                    singpast2(marketed),
                    singpast3(marketed),
                    past(marketed),
                    pl(market),
                    sing1(market),
                    sing2(market),
                    sing3(markets)])]),
   qualia([formal([sell]),
          telic([]),
          const([]),
          agent([])]),
   args([arg1(A1,
             syn([type(np)]),
             qualia([formal([organization])])),
        arg2(A2,
             syn([type(np)]),
             qualia([formal([device])]))]),
   cospec([
          [A1,*,self,*,A2],
          [A1,*,in,*,A2,self],
          [self,*,A2],
          [A2,*,self]
         ]),
   types(capability_verb),
   template_semantics(prim_cap([purch(A1)],A2))).
```

The two arguments to the verb *market* are minimally typed as a result of the conversion from LDOCE, this information being represented as a *type-path* for each argument, Pustejovsky and Boguraev[12]. For example, the subject is typed as a **organization** and the object as the type **device**.

The syntactic and collocational behavior of a word is represented in the **cospec** (cospecification) field of the entry. The **cospec** of a lexical item can be seen as paradigmatic syntactic behavior for a word, involving both abstract types as well as lexical collocations. This field is created automatically by reference to the syntactic codes of the verb, as represented in LDOCE, in this case T1 (i.e., basic transitive). That is, the **cospec** encodes explicit information regarding the linear positioning of arguments, as well as semantic constraints on the arguments as imposed by the typing information in the qualia. The syntactic representation of a word's environment may appear flat, but the semantic interpretation is based on a unification-like algorithm which creates a much richer functional structure. Theoretically, the expressive power of converting the **cospecs** of a GLS into DCG parse rules is equivalent to the power of a Lexicalized Tree Adjoining Grammar with collocations (Shieber[14]), what we have termed Hyper Lexicalized Tree Adjoining Grammars (HTAGs) (Pustejovsky[13]).

Lexically Encoding Idiomatic and Phrasal Structures

One of the advantages to the highly lexical approach being taken here is the ability to encode idiomatic expressions and phrasal expressions as part of the lexicon proper, where motivated by statistical confirmation of the collocations. For example, in the English JV corpus, it so happens that reporting verbs such as *announce* very often appear with tensed clause complements carrying pronominal subjects, as below:

IBM$_i$ announced that it$_i$ had entered a joint venture with Mark Computers.

The standard linguistic approach to resolving the coreference between the pronoun and the company name is to syntactically "walk" the parse tree and bind the pronoun according to standard syntactic anaphora constraints. That is, the subject of the matrix sentence is a ***possible antecedent*** to the pronoun in the lower clause. Within a lexical approach, such stereotypical antecedent-anaphora pairs such as that above are directly encoded in the **cospec** of the appropriate lexical trigger. For example, the cospec for such a pattern with the verb *announce* is given below:

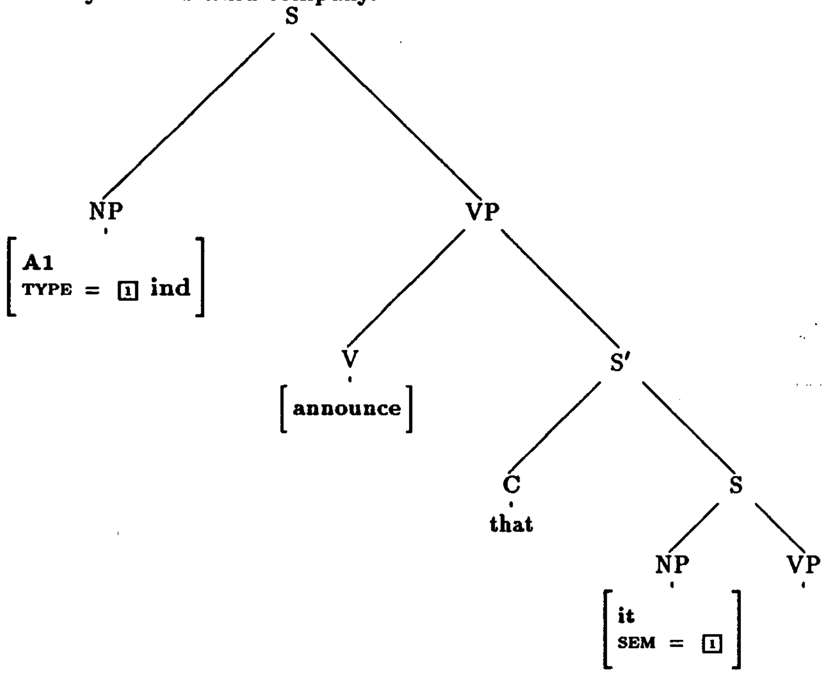
```
gls(announce,
   syn([type(v),
   ....
   args([arg1(A1,
             syn([type(np)]),
             qualia([formal([organization])])),
   ....])]),
   cospec([....,
           [A1,self,that,it,*,A2],
         ]),
   template_semantics()).
```

This corresponds to the tree fragment of a Hyper-lexicalized TAG shown on the next page.

The ability to lexically encode domain- or sublanguage-specific idioms or phrases is also useful for referring (i.e., naming) expressions. For example, in the JV domain, a phrase such as

the new company, called COMPANY-NAME

is triggered by the lexical item *called*, with the cospecification constraint of being an anaphoric reference to the newly created child company.



Creation of a Lexically-Driven Partial Parser

The generative lexical structures are "universal" in character in the same way that phrase structural descriptions provide a general, expressive language for describing the syntactic structures for widely different linguistic behaviors from different languages. The lexical entries for Japanese follow exactly the same specifications, with the same degree of flexibility.

The partial grammar is derived from the tuned GLS entries. Prolog Definite Clause Grammar rules are produced automatically from the patterns given in the GLS cospecification. The rules are then compiled into the working version of the system. A similar process also produces lists of words with GLS entries for the tagging programs.

For example this partial GLS entry will give rise to two parse rules.

```
gls(join,
   syn([type(v),
        code(gcode_t1),
        eventstr([]),
        ldoce_id(join_1_2),
...
         arg3(A3,
              syn([type(np)]),
              qualia([formal([organization]),
                      telic([]),
                      const([]),
                      agent([])]))]),
   cospec([[A1,*,self,*,A2,*,with,A3],
          [A1,self,A3,to,*,A2],
          [A1,A3,*,self,*,A2],
          [A1,together,with,A3,*,self,*,A2]]),
   types(tie_up_verb),
   template_semantics(prim_tie_up([A1,A3],A2,[f(tie_up_status,existing,[])]))).
```

The single pattern - [A1,*,self,*,A2,*,with,A3] gives the rule -

```
rule(join,
 template_semantics(prim_tie_up([A1,A3],A2,,[f(tie_up_status,existing,[])])))
  -->
     glsphrase(A1,[type(np)],formal([organization])),
     ignore,
     term(gls(_,type([join,v]),_)),
     ignore,
     glsphrase(A2,[type(np)],formal([organization])),
     ignore,
     word(with),
     glsphrase(A3,[type(np)],formal([organization])).
```

Specific Issues – EJV Text 0592

(1) Coreference determination

```
   for:  "LOCAL CONCERN",
 "UNION PRECISION CASTING CO. OF  TAIWAN"
```

We do not have a mechanism to associate "IN TAIWAN" and "LOCAL CONCERN" we additionally failed to identify the company name completely.

```
   for:  "A  JAPANESE  TRADING HOUSE",
         "TAGA CO., A COMPANY ACTIVE IN TRADING WITH  TAIWAN"
```

We did not have any lexical entry for "TRADING HOUSE", when this was added we detected a Japanese company. However, it is not at all clear how we determine that "TAGA" is the company. The reference depends on three companies being mentioned and three referred too.

```
   for:  "A  JOINT VENTURE",
         "THE NEW COMPANY, BASED IN KAOHSIUNG, SOUTHERN  TAIWAN",
```

We missed the location – couldn't tie "new company" to anything.

```
         "THE JOINT VENTURE,  BRIDGESTONE SPORTS  TAIWAN  CO.",
```

The noun phrase marks the company as the child and this is resolved back to the empty child slot from the previous sentence.

```
         "THE  TAIWAN  UNIT"
```

Missed this as we have no lexical entry for "UNIT".

```
   for:  "BRIDGESTONE SPORTS CO.",
```

We get this first reference, then throw it away in the inference stage (not enough evidence from the parse – one parent name, no child name or partner

```
         "BRIDGESTON SPORTS",
```

This ref. appears in Para 2, naming an announcement. We miss it.

```
         "BRIDGESTONE SPORTS",
```

This ref is the in the ownership, and we get it here. It's then resolved across to the tie-up.

"THE JAPANESE SPORTS GOODS MAKER"

Once again, no lexical entry for "MAKER"

(1a) Which coreferences did your system get? Of those, which could it have gotten 6 months ago (at the previous evaluation)? How can you improve the system to get the rest?

At the 18 month, we got 2 separate tie-ups for two different mentions of a jv-like event. Our system is more precise now and we need to improve recall. The ownership patterns, which are essentially lists of pairs, need an additional mechanism, or a new semantic type (COMPANY+PERCENT) to ensure their detection.

(2) Did your system get the OWNERSHIPs, in particular from "... THE REMAINDER BY TAGA CO."?

The quote from the article is:

```
    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.
```

We get the 75 percent, and none of the others.

Score: Precision 94, Recall 58, P&R 72 (after fix to lexicon)

Specific Issues – EME Text 2789568

(1) What information triggers the instantiation of each of the two LITHOGRAPHY objects?

"Stepper" has semantic tags which indicate it participates in a LITHOGRAPHY object -

```
equip('stepper',type([[process_type,'LITHOGRAPHY'],[equipment_type,'STEPPER']]))
```

(2) What information indicates the role of the Nikon Corp. for each Microelectronics Capability? The GLS patterns – `<Organization> ... market ... <Equipment>` gives us manufacturer and distributor.

(3) Explain how your system captured the GRANULARITY information for "The company's latest stepper."

Didn't get either one, even though the measures are tagged, and the word "resolution" is marked as a process identifier. The parser co-ordination mechanism split the granularity measures into a separate process list. This did not get resolved, as it had no process type or other information for the resolver.

4) How does your system determine EQUIPMENT_TYPE for "the new stepper"? and for "the company's latest stepper"?

They are "stepper"s – there is a lexical item for this.

(5) How does your system determine the STATUS of each equipment object?

This is supplied by the GLS entry. The parse, or the inference stage in some cases, detects the entity role - here it is "MANUFACTURER" - and infers that the equipment is in use. Entity roles are also inferred from roles attached to particular nouns (e.g. University = Developer).

(6) Why is the DEVICE object only instantiated for LITHOGRAPHY-1?

The reference resolver will always resolve devices to the closest process type. Here we have only produced only one anyway.

Score: Precision 83, Recall 34, P&R 49

Specific Issues – JJV Text 0002

(1) How to detect a reportable tie-up

The GLS verb and the cospec patterns in the text determine whether or not there is a reportable tie-up in the text.

(2) How many tie-ups in article 0002 and strategies to detect the second tie-up in the article

The system detects three tie-ups. The tie-up in the second sentence is captured based on a GLS verb ('提携' teikei = tie up) in the sentence and its copec patterns, and there is only one tie up in the sentence and all the detected organizations are treated as partners.

(3) How to detect entities in a tie-up

At the sentence level, the cospec patterns of GLS verbs determine the number of entities in the tie-up. In addition, within the sentence the tags (for organization, person) and local context are used by the parser to decide the entities in the tie-up.

(4) The number of discourse entities and how to determine whether they are reportable

There are four entities, three tie-ups, two industries and two activities. The entities and tie-ups are determined by the GLS cospec patterns which have been transformed into grammar rules.

(5) Difficulties in identifying the correct number of entities, tie-ups, and tie-up relations

The entities and tie-ups are determined by the GLS cospec patterns which have been transformed into grammar rules. As the recall goes up, the system captures more objects and the resolver should resolve them correctly. If the resolver does so too loosely and resolve many objects into a few, it is the case of under-generation. If the resolver does it too strictly, then it is over-generation. The third tie-up for article 0002 should be resolved with the first tie-up.

(6) How to detect aliases

The system uses character-based rules for identifying aliases. For example, if a company name starts with '日立' (Hitachi) as in '日立製作所' (Hitachi Manufacturing), then the system looks for the string '日立' (Hitachi) or the first two characters of the company name as its alias. The system also stores special aliases in its knowledge base which are difficult to recognize with the character-based rules, for example, 'ＪＡＬ' for '日本航空' (Japan Airlines).

(7) Problems in detecting the alias for the ENTITY named Toukyou Kaijou Kasai Hoken

Both the full name and the alias of the ENTITY are tagged as organization at the tagging stage and they are resolved as the same organization by the resolver later on in the system. Thus the system correctly recognized the alias for the ENTITY.

(8) How to decide a general description of an activity in the second sentence of article 0002

Whether a description is general or not, if there is an activity term together with a verb which shows some on-going economic activity in a sentence, they will produce an activity object.

(9) The way the system handles 'ryousha' (the both companies) in sentence 2 in 0002 and the particle 'の' (no = of)

The system treats 'ryousha' as legitimate organization without a specific name. The reference resolver tries to resolve 'ryousha' with two particular organizations. However, in the case of an organization name followed by 'no ryousha' ('の両社') is commented out at the tagging stage because it does not give any more specific information than two particular organization names which precedes it, and thus ignored by the following processes in the system.

Score: Precision 72, Recall 72, P&R 72

Specific Issues – JME Text 0452

(1) How to determine the existence of a reportable microelectronics capability.

The GLS verb and the cospec patterns in the text determine whether or not there is a reportable microelectronics capability in the text.

(2) Three entities are mentioned in the article. How does the system determine which were involved in the ME capability?

At the sentence level, the cospec patterns of GLS verbs determine the number of entities in the ME capability. The tie-up company is not captured because of the lack of the coverage of current GLS verbs.

(3) How to identify company names and how to associate them with their locations.

Companies or organizations are detected in two ways. First for known company names are tagged by straightforward string matching using a list of company names. Second, for unknown companies, the system tries to find them by using particle information and local context in the sentence. The two company names in the first sentence are identified because they are in the list of known organization names. If the location information is located close to the company name, it is grouped with the company name and treated as a part of the company information by the system.

(4) How to associate film type with each ME capability.

If the film type information is located close to the detected process, then it is associated with the process as in article 0452. If the film type is found remotely, then reference resolver tries to associate it with a proper process if found.

(5) How to determine the existence of reportable equipment.

An equipment name is tagged at the tagging stage using a list of known equipment names such as CVD system, PVD system. If a process is followed by specific terms which indicate clearly an equipment, such as '装置' (souchi = equipment or system), then the process and the term are changed into an equipment with the process information included. At the parsing stage, if the equipment is located in the sentence in such a way that it matches with one of the cospec patterns of the GLS verb in the sentence, then it is detected as reportable equipment. It is usually associated with one or more organizations. Detection of a stand alone equipment does not by itself generate a new ME capability.

Score: Precision 91, Recall 72, P&R 81

CONCLUSIONS

We have learned a great deal over the past two years, partly through the many mistakes we have made. The project has depended a great deal on the skill and care of the people working on it to ensure consistency in our data and code. Given the large number of knowledge bases in our system this is an onerous task and one task needed for the future is a system which allows this knowledge to be integrated and held in one central data-base, where consistency can be maintained. The second is to develop an easily configurable and portable reference resolution engine.

There are no major differences in the structure of the English and Japanese systems. It would seem that a critical part of achieving high precision and recall is to have at least one person with a reasonable knowledge of the whole system to carry out repeated test/improve cycles.

The current system is robust and provides a good starting point for the application of more sophisticated techniques, some of them simply refined versions of the current architecture. Given appropriate data it should be possible to produce a similar system for a different domain in a matter of months. Many parts of the system are portable in particular the semantic tagging mechanisms, the statistical filtering component. Dates, companies and people - all of which occur in many kinds of text - we now handle with good levels of accuracy. The strange conventions of equipment names have provided us with some interesting new challenges.

ACKNOWLEDGEMENTS

The system described here has been funded by DARPA under contract number MDA904-91-C-9328.

We would like to express our thanks to our colleagues at BBN who have shared their part of speech tagger (POST) with us. Thanks also to Kyoto University for allowing us to use the JUMAN segementor and part of speech tagger.

Diderot is a team effort and is a result of the work of many people. The following colleagues at CRL and Brandeis have contributed time, ideas, programming ability and enthusiasm to the development of the Diderot system; Paul Buitellar, Federica Busa, Peter Dilworth, Steve Helmreich and Fang Lin.

References

[1] DARPA. *Proceedings of the Third Message Understanding Conference (MUC-3)*, San Mateo, CA, 1991. Morgan Kaufmann.

[2] DARPA. *Proceedings of the Fourth Message Understanding Conference (MUC-4)*, San Mateo, CA, 1992. Morgan Kaufmann.

[3] Cowie, J., Guthrie, L., Wakao, T., Jin, W., Pustejovsky, J. and Waterman, S., The Diderot Information Extraction System. In *Proceedings of the First Conference of the Pacific Association for Computational Linguistics (PACLING93)*, Vancouver, Canada, 1993.

[4] Grishman, R., and Sterling, J., Acquisition of selectional patterns. In *Proceedings of the 14th International Conference on Computational Linguistics (COLING92)*, Nantes, France, 1992.

[5] Guthrie, L., Bruce, R., Stein, G.C., and Weng, F., Development of an application independent lexicon: Lexbase. Technical Report MCCS-92-247, Computing Research Laboratory, New Mexico State University, 1992.

[6] Guthrie, L., and Walker, E., Some comments on classification by machine. Technical Report MCCS-92-935, Computing Research Laboratory, New Mexico State University, 1992.

[7] Lehnert, W., and Sundheim, B., An evaluation of text analysis technologies. *AI Magazine*, 12(3):81–94, 1991.

[8] Proctor, P., editor. *Longman Dictionary of Contemporary English*. Longman, Harlow, 1978.

[9] Pustejovsky, J., The generative lexicon. *Computational Linguistics*, 17(4), 1991.

[10] Pustejovsky, J., The acquisition of lexical semantic knowledge from large corpora. In *Proceedings of the DARPA Spoken and Written Language Workshop*. Morgan Kaufmann, 1992.

[11] Pustejovsky, J., Bergler, S., and Anick, P., Lexical semantic techniques for corpus analysis. *Computational Linguistics*, 1993.

[12] Pustejovsky, J. and Boguraev, B., Lexical knowledge representation and natural language processing. *Artificial Intelligence*, 1993.

[13] Pustejovsky, J., *The Generative Lexicon: A Computational Theory of Lexical Semantics* MIT Press, Cambridge, MA, 1994.

[14] Schabes, Y. and Shieber, S., An alternative conception of tree-adjoining derivation. In *Proceedings of 30th Annual Meeting of the Association for Computational Linguistics*, 1992.

Summary of Error-based Scores

JAPANESE MICRO

	ERR	UND	OVG	SUB	Min	Max
18-Month	72	60	28	18	.74	.80
24-Month	65	54	24	12	.69	.73

JAPANESE JV

	ERR	UND	OVG	SUB	Min	Max
18-Month	79	71	22	22	.86	.86
24-Month	63	51	23	12	.70	.72

ENGLISH MICRO

	ERR	UND	OVG	SUB	Min	Max
18-Month	86	76	33	37	.87	.93
24-Month	74	60	33	24	.80	.84

ENGLISH JV

	ERR	UND	OVG	SUB	Min	Max
18-Month	91	76	40	56	1.06	1.08
24-Month	79	67	28	28	0.89	0.91

Summary of Recall/Precision-based Scores

JAPANESE MICRO

	TF(R/P)	REC	PRE	P & R
18 - Month	73/83	32	59	41.99
24 - Month	84/90	40	66	50.37

JAPANESE JV

	TF(R/P)	REC	PRE	P & R
18 - Month	82/99	26	61	32.8
24 - Month	88/98	42	67	52.1

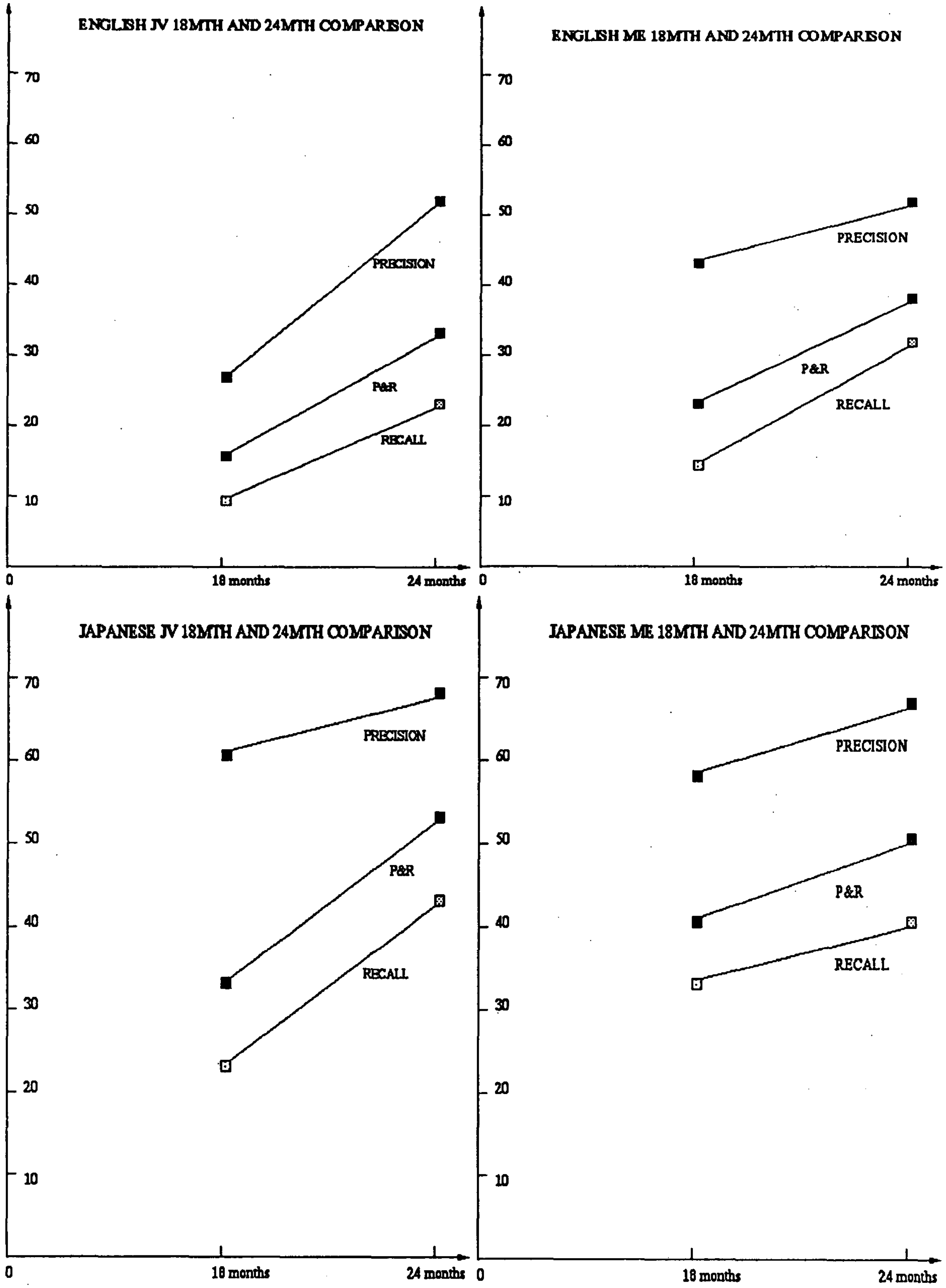
ENGLISH MICRO

	TF(R/P)	REC	PRE	P & R
18 - Month	77/76	15	42	22.28
24 - Month	78/83	31	51	38.49

ENGLISH JV

	TF(R/P)	REC	PRE	P & R
18 - Month	67/86	10	26	15.10
24 - Month	76/92	24	51	32.64

Progress since 18 month workshop



ENGLISH JV 18MTH AND 24MTH COMPARISON

ENGLISH ME 18MTH AND 24MTH COMPARISON

JAPANESE JV 18MTH AND 24MTH COMPARISON

JAPANESE ME 18MTH AND 24MTH COMPARISON