UNIVERSITY OF SHEFFIELD: DESCRIPTION OF THE LaSIE SYSTEM AS USED FOR MUC-6

R. Gaizauskas, T. Wakao, K. Humphreys, H. Cunningham, Y. Wilks
Department of Computer Science, University of Sheffield
{robertg,wakao,kwh,hamish,yorick}@dcs.shef.ac.uk

Introduction and Background

The LaSIE (Large Scale Information Extraction) system has been developed at the University of Sheffield as part of an ongoing research effort into information extraction and, more generally, natural language engineering.

LaSIE is a single, integrated system that builds up a unified model of a text which is then used to produce outputs for all four of the MUC-6 tasks. Of course this model may also be used for other purposes aside from MUC-6 results generation, for example we currently generate natural language summaries of the MUC-6 scenario results.

Put most broadly, and superficially, our approach involves compositionally constructing semantic representations of individual sentences in a text according to semantic rules attached to phrase structure constituents which have been obtained by syntactic parsing using a corpus-derived context-free grammar. The semantic representations of successive sentences are then integrated into a 'discourse model' which, once the entire text has been processed, may be viewed as a specialisation of a general world model with which the system sets out to process each text.

LaSIE has a historical connection with the University of Sussex MUC-5 system [GCE93] from which it derives its approach to world modelling and coreference resolution and its approach to recombining fragmented semantic representations which result from partial grammatical coverage. However, the parser and grammar differ significantly from those used in the Sussex system. In its approach to named entity identification LaSIE borrows to some extent from the approach adopted in the MUC-5 Diderot system [CGJ$^+$93]. Virtually all of the code in LaSIE is new and has been developed since January 1995 with about 20 person-months of effort.

System Description

Significant Features

LaSIE is an integrated system, performing lexical, syntactic and semantic analysis to build a single, rich representation of the text which is then used to produce the MUC-6 results for all four tasks. Features which distinguish the system are:

- an integrated approach allowing knowledge at several linguistic levels to be applied to each MUC-6 task (e.g. coreference information is used in named entity recognition);
- the absence of any overt lexicon – lexical information needed for parsing is computed dynamically through part-of-speech-tagging and morphological analysis;
- the use of a grammar derived semi-automatically from the Penn TreeBank corpus;
- the use and acquisition of a world model, in particular for the coreference and scenario tasks;
- a summarisation module which produces a brief natural language summary of scenario events.

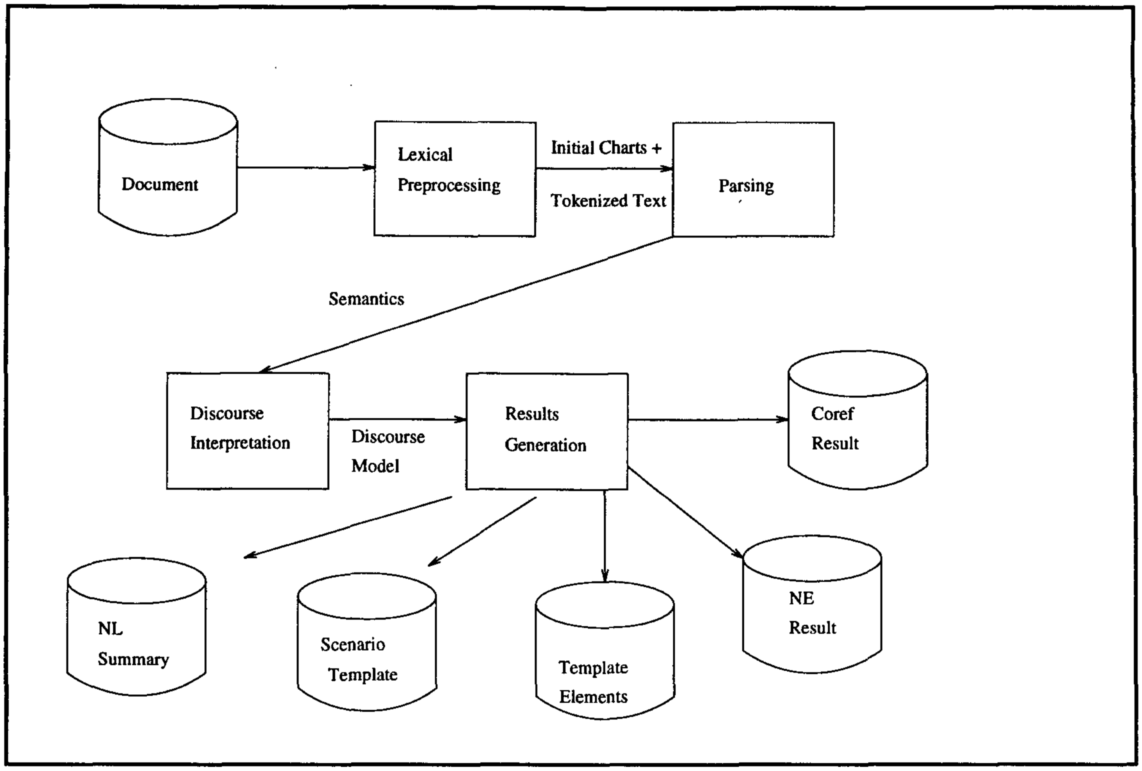

Figure 1: LaSIE System Architecture

Architecture Overview

The high level structure of LaSIE is illustrated in Figure 1. The system is a pipelined architecture consisting of three principle processing stages: lexical preprocessing, parsing plus semantic interpretation, and discourse interpretation. Note that none of these high level system components corresponds directly to any MUC-6 task. This reflects the fact that LaSIE has been designed as a general purpose information extraction research system, initially geared towards, but not solely restricted to, MUC-6 tasks. Further, note that all MUC-6 results are generated after the most complete text representation has been built. This reflects the desire to use information derived from all levels of linguistic processing in performing each of the MUC-6 tasks.

Lexical Preprocessing

Input to the lexical preprocessor is a flat ASCII file containing a single Wall Street Journal article marked up to the paragraph level in SGML. Output consists of two parts: a sequence of lexically seeded charts for input to the parser and a byte-offset/token representation of the initial text for later reconstruction with added markup in the results module.

Processing consists of tokenising and sentence-splitting the input, part-of-speech tagging the tokens, performing morphological analysis to obtain root forms, pattern-matching against precompiled lists of named entities, and finally the creation of lexical or phrasal feature-structured edges for input to the parser.

Note that no conventional lexicon is used; dynamic tagging and morphological analysis provide all of the information required in the parser.

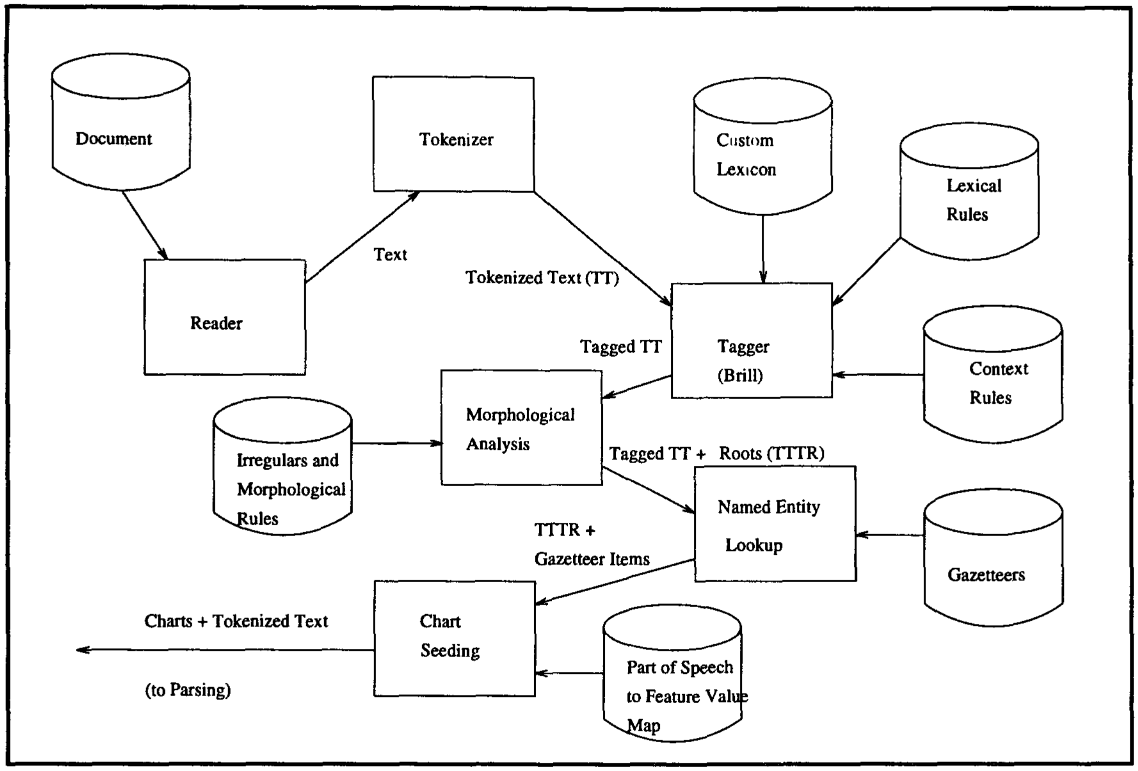

Figure 2: Lexical Preprocessing

The Tokenizer

Tokenization serves two purposes. Firstly token boundaries (as byte offsets into the text) are identified and each token is assigned an identifier (its number in the sequence that makes up the document). These identifiers are preserved throughout the system and used with the associated offsets to facilitate allocation of SGML markup for NE and coreference annotation to byte positions in the MUC-6 results files.

Secondly, since the Brill tagger expects one sentence per line and expects each token to be separated by white space, the text stream is changed into this format before it is fed to the tagger. BBN's POST program [WMS+93] is run separately on the text as a sentence splitter to provide the sentence boundary information.

Part-of-Speech Tagging

The Brill tagger [Bri94] is a rule-based part-of-speech tagger that has been extensively trained on Wall Street Journal Text. It uses the Penn Treebank tag set which consists of 48 part-of-speech tags [MSM93]. We have custom-configured the tagger in a number of ways. These changes include introducing new tags for dates, SGML markup, and for several punctuation symbols that are treated identically by the default tagger. We have also added several lexical and contextual rules to the tagger's rule base.

Here is a sample from the walkthrough text of the transformed input stream after tokenizing and tagging. Each line corresponds to a single token and gives paragraph number, sentence number within paragraph, start byte-offset, end byte-offset, token, and part-of-speech tag of the token.

```
16 1 483 485 <p> SGML
17 1 490 492 One CD
17 2 494 495 of IN
17 3 497 499 the DT
17 4 501 504 many JJ
17 5 506 516 differences NNS
17 6 518 524 between IN
```

```
17 7 526 531 Robert NNP
17 8 533 533 L NNP
17 9 534 534 . PERIOD
```

Morphological Analysis

Following tagging, all nouns and verbs are passed to a morphological analyser, which returns a root form for inclusion in the initial parser input. A set of 34 regular expression rules performs the analysis, in conjunction with a list of around 3000 irregular exceptions derived semi-automatically from the exception list used in WordNet [Mil90]. The morphological analyser is implemented by compiling the irregulars and rules into a *flex* program which is translated to C.

Named Entity Phrasal Tagging

Before parsing an attempt is made to identify and to tag named entity related phrases. This is done both by matching the input against pre-stored lists of proper names, date forms, currency names, etc. and by matching against lists of common nouns that act as reliable indicators or triggers for classes of named entity. These lists are compiled via *flex* program into a finite state recogniser. Each sentence is fed to the recogniser and all single and multi-word matches are used to associate token identifiers with named entity tags.

Lists of names are employed for locations, personal titles, organizations, dates/times and currencies. The following lists of names are used.

- Organization names: a cleaned up list originally from the Consortium for Lexical Research (CLR) anonymous ftp site, containing about 2600 names.
- Company Designator: 'Co.','Ltd','PLC', etc. – 94 designators based on the company designator list provided in the MUC6 reference resources.
- Titles: 'President', 'Mr.' – about 160 titles, manually collected.
- Human names: mainly first names, numbering about 500, based on a list of names in the Oxford Advanced Learner's Dictionary [Hor80].
- Currency units: e.g. 'dollars', 'pounds', etc. – 101 such unit names, taken from the MUC6 reference resources.
- Location names: names of major cities in the world as well as province/state and country names. Derived from a gazetteer list of about 150,000 place names by taking the highest level ('level 1') entries only – 225 country, 1189 province, and 854 city names in total.
- Time expressions: phrases like 'first quarter of' – 49 phrases, manually constructed.

A trigger word is a word which indicates that the tokens surrounding it are probably a named entity item and may reliably permit the type or even subtype of the named entity to be determined (e.g. company and government are subtypes of type organization). The lists of trigger words were produced by hand.

- Location: 8 trigger words for location names, e.g. 'Gulf','Mountain'.
- Organization:
  - Government institutions: 7 trigger words for governmental institutions, e.g. 'Agency', 'Ministry'.
  - Company: 138 trigger words for companies, e.g. 'Airline', 'Association'.

The above names and key words are specially tagged as result of the list lookup stage, and are used in Named Entity grammar rules.

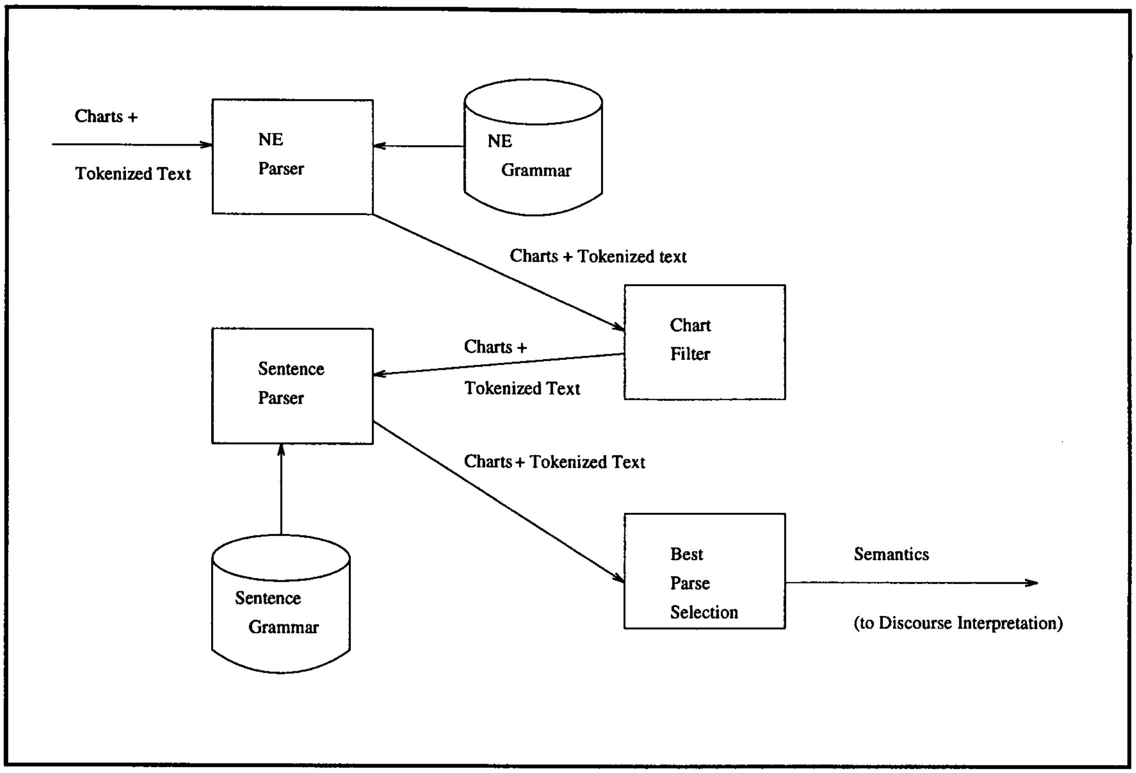

Figure 3: Parsing and Semantic Interpretation

Chart Seeding

For each sentence an initial chart is constructed which contains an edge for each lexical item and list-matched named entity item in the sentence. The edges contain feature-value structures holding information about the items. The feature values are filled in from the information already present in the data stream (POS tags, results of morphological analysis). A POS-tag-to-feature-value map database is used to associate one or more feature-values with POS tags (e.g. the Penn Treebank tag NNS maps on to the features `cat:noun, number:sing, person:3`).

Parsing

The LaSIE parser is a simple bottom-up chart parser implemented in Prolog. The grammars it processes are unification-style feature-based context free grammars. During parsing semantic representations of constituents are constructed entirely using Prolog term unification. When parsing ceases, i.e. when the parser can generate no further edges, a 'best parse selection' algorithm is run on the final chart to chose a single analysis. The semantics are then extracted from this analysis and passed on to the discourse interpreter.

Parsing takes place in two passes, each using a separate grammar. In the first pass a special named entity grammar is used, the sole purpose of which is to identify noun phrases relevant to the MUC-6 named entity task. These constituents are then treated as unanalyzable during the second pass which uses a more general 'sentence' grammar.

Named Entity Grammar

The grammar rules for Named Entity items constitute a subset of the system's noun phrase (NP) rules. All the rules were produced by hand. There are 206 such rules in total of which 94 are for organization, 54 for person, 11 for location, 18 for date/time, and 29 for money/percent expressions.

Grammar rule examples

Here are some examples of the Named Entity grammar rules:

```
NP --> ORGAN_NP
ORGAN_NP --> LIST_LOC_NP NAMES_NP CDG_NP
ORGAN_NP --> LIST_ORGAN_NP NAMES_NP CDG_NP
ORGAN_NP --> NAMES_NP '&' NAMES_NP
NAMES_NP --> NNP NAMES_NP
NAMES_NP --> NNP PUNC(_) NNP
NAMES_NP --> NNP
```

The non-terminals LIST_LOC_NP, LIST_ORGAN_NP and CDG_NP are tags assigned to one or more input tokens in the NE phrasal tagging stage of lexical preprocessing. The non-terminal NNP is the tag for proper name assigned to a single token by the Brill tagger.

The rule ORGAN_NP --> NAMES_NP '&' NAMES_NP means that if an as yet unclassified or ambiguous proper name (NAMES_NP) is followed by '&' and another ambiguous proper name, then it is an organization name. An example of this is "Ammirati & Puris" in the walkthrough text, which matches this pattern and is therefore classified as an organization following NE parsing.

Nearly half of the NE rules are for organization names because they may contain any other proper names (such as personal names, location names) as well as normal nouns, and their combinations. There are also a good number of rules for personal names since care needs to be taken of first names, family names, titles (e.g. 'Mr.','President'), and special lexical items such as 'de' (as in 'J. Ignacio Lopez de Arriortua') and 'Jr.','II', etc.

There are not so many rules for location names because they are recognized mainly in the previous preprocessing stage by looking them up in the lists of city, province/state, country, and region names.

Rules for monetary and time expressions have been collected by analysing actual expressions in the training texts.

Sentence Grammar Rules

The grammar used for parsing at the sentence level was derived from the Penn TreeBank-II (PTB-II) [MSM93], [MKM+95]. Since the PTB-II contains a large skeletally parsed corpus of Wall Street Journal articles, it seemed to us worth investigating as a potential source of a grammar for the MUC-6 tasks. Research into number and frequency distribution of rules in this corpus led to some surprising findings [Gai95a]. If a number of simplifying assumptions are made, a context-free grammar can be extracted from the PTB-II WSJ corpus. Doing so led to an unmanageably large grammar: approximately 17,500 rules. However, only a small number rules account for the majority of rule occurrences. The following table illustrates the number of rules in a grammar which accounts for the top $n\%$ of rule occurrences for each category which is either an S or occurs as a nonlexical category on the right hand side of some other rule included in the grammar:

% Rule Occurrences	Grammar Size in Rules
100	17540
95	2144
90	872
80	240
70	112

Given the speed of our parser, the repair mechanisms for fragmentary parses in later parts of the system, and the difficulty of manually assigning semantic rules to large numbers of syntactic rules, we opted for the 112 rule grammar representing 70 % of rule occurrences of the principal PTB constituent categories.

When parsing for a sentence is complete the resultant chart is analyzed to extract the 'best parse'. Our algorithm for this was as follows: identify the set of syntactic categories for which useful standalone semantics can be assigned – in our case S, NP, VP, and PP. Extract the set of shortest sequences of maximally spanning, non-overlapping edges of these categories. In the event of this set containing more than one member, pick one arbitrarily and designate it the 'best parse'. From the 'best parse' the associated semantics are extracted to be passed on to the discourse interpreter.

Semantic structures were assigned by hand to the set of rules automatically derived from the PTB-II corpus. For simple verbs and nouns the morphological root is used as a predicate name in the semantics, and tense and number features are translated directly into the semantic representation where appropriate.

All NPs and VPs lead to the introduction of a unique instance constant in the semantics which serves as an identifier for the object or event referred to in the text – e.g. *company* will map to something like `company(e22)` in the semantics and *hired* to `hire(e34)`, `time(e34,past)`. Each of these instance constants is given a `realisation` property in the semantic representation, indicating, as a token range, the position in the text from which the semantics were derived. Nouns used as possessives or qualifiers also require realisation properties, necessitating the introduction of these categories into the grammar, although they did not occur in the original PTB-II ruleset. Each instance in the semantics is also augmented with further `realisation` properties specifying the sentence number and paragraph number in which the instance's surface token range occurs, and also whether the range is part of the header or the body of the article. These `realisation` properties provide back pointers from the semantics into the text and are necessary for writing out coreference markup and doing summarisation (they permit original surface forms to be used in the summaries). This requirement to go back from semantic representation to surface text was one of the biggest innovations in our system that MUC-6 required.

A small set of hand constructed rules were also used in addition to those automatically derived from PTB-II, to extend the coverage of the grammar for particular constructions such as possessives. These additional rules are also used to combine verb-particle sequences into a compound form for use as a predicate name in the semantics.

For example, the phrase "stepping down as chief executive officer" will be represented in the following form:

```
step_down(e58),
time(e58,present),
realisation(e58,tokens(225,230)),
realisation(e58,sentence(10)), realisation(e58,section(4)), realisation(e58,type(body)),
as(e58,e60),
title(e60,'chief executive officer'),
realisation(e60,tokens(228,230)),
realisation(e60,sentence(10)), realisation(e60,section(4)), realisation(e60,type(body))
```

Discourse Interpretation

The Discourse Interpreter module translates the semantic representation produced by the parser into a representation of instances, their ontological classes and their properties, using the XI knowledge representation language [Gai95b]. XI allows a straightforward definition of cross-classification hierarchies and the association of arbitrary properties with classes or instances. These properties may be simple attribute-value associations, such as `name(e1,'PaineWebber')` or `animate(person(X),yes)`, or they may be rules which specify how the value of an attribute is to be derived from other information in the hierarchy, e.g.:

> `org_locale(e1,e2)` if `in(e1,e2)` and `e2` is an instance of the class `location`

XI provides a simple inheritance mechanism which allows properties to be inherited by classes or instances lower in the hierarchy.

Ontology

The definition of a cross-classification hierarchy is referred to as an *ontology*, and this together with an association of attributes with nodes in the ontology forms a *world model*. The basic ontology used for the MUC-6 tasks is extremely simple, consisting of only 40 predefined object classes and 46 attribute types. For the scenario task, a hierarchy of 39 event classes and 9 additional attribute types were added. The

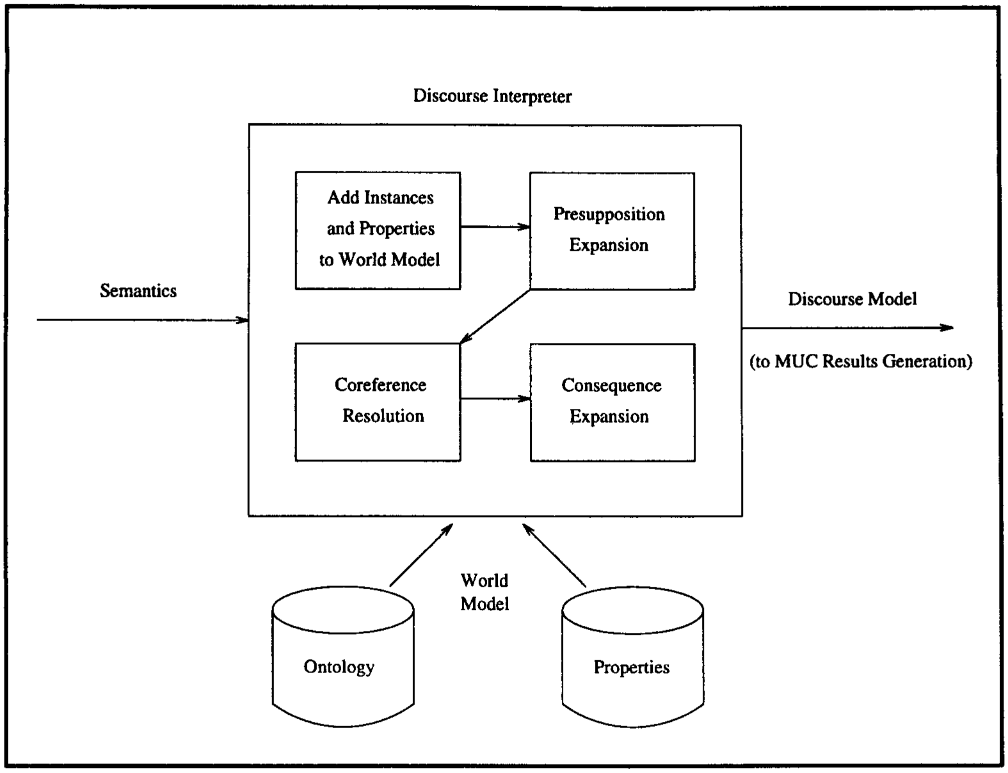

Figure 4: Discourse Interpretation

ontology, together with the attributes associated with the nodes, were produced manually with the classes and attributes being motivated mainly by the requirements of the MUC-6 tasks. During the processing of a text, new classes of objects and events are automatically added to the ontology to enrich the hierarchy. The new nodes are currently only added as direct subclasses of either objects or events and so the number of inheritable properties is extremely limited, but this mechanism does allow coreferences between instances of a class previously unknown in the ontology, for instance two mentions of "sailing" in the walk-through article.

As well as having attribute with atomic values, a node in the ontology may have attributes whose values are specified by inference rules associated with it. The addition of an instance or property of a certain class to the world model while processing the text will lead to the evaluation of any inherited inference rules, potentially causing the addition of further instances or properties to the world model, or the reclassification of existing instances. For example, the addition of `name(e1,'PaineWebber')` will cause the addition of `e1` as an instance of the `object` class, via the rule associated with the `name` property type which states that only objects can have name properties. We refer to this as *presupposition expansion.* A similar set of scenario specific inference rules are evaluated following coreference resolution in the consequence expansion stage.

Coreference Resolution

The semantic representation of a text is added to the world model sentence by sentence, with any presupposition expansion carried out immediately. After each sentence, all newly added instances are compared with previously added instances to determine whether any pair can be merged into a single instance, representing a coreference in the text. The comparison of instances is carried out in several stages, terminating if a match is found during any one stage, as follows:

1. new instances with `name` properties are compared with all existing instances with `name` properties, i.e. named entity coreferences can range over the whole text;
2. all new instances are compared with each other (intrasentential coreference resolution);

3. new pronoun instances are compared with existing instances from the same paragraph as the current input sentence, i.e. pronoun coreferences are intra-paragraph only, with an exception for paragraph initial pronouns to allow reference to the previous paragraph;
4. all other new instances are compared with existing instances from the current and previous paragraphs, i.e. all other coreferences are restricted to a span of two paragraphs.

Each comparison involves first determining if there is a path between the instances' classes in the ontology. If no such path exists then the instances are on different branches of the ontology, and so such pairs are not further considered for coreference. If a path does exist then the attributes of the instances are compared to ensure no conflicts exist. Certain attributes, such as `animate` are defined as having unique fixed values for any instance and so instances with conflicting values for these attributes cannot be the same. If such conflicts are discovered then the comparison is abandoned. The `name` attribute is treated specially, using a semantic type specific name match (described below) to determine the compatibility of the newly input instance's name with the longest known name of the existing instance.

If no attribute conflicts are found between two instances, a similarity score is calculated based on the number of common properties and on a semantic distance measure, determined simply in terms of the number of nodes in the path between them. After a newly input instance has been compared with all others in a particular comparison set, it is merged in the world model with the instance with the highest similarity score, if one exists. In the case of equal scores for two or more previous instances, which is common in the case of pronouns in the input, the most recent comparison, corresponding to the closest pair in the text, is preferred.

Name Matching

Coreference resolution for Named Entity items (proper names) is important in order to recognize various forms of proper names, especially organization names. For example, 'Ford Motor Co.' may be used in a given text the first time the company is mentioned, and subsequently it may be referred to as 'Ford'.

In order to determine whether given two proper names (organization, person, location) match or not, various heuristics are used, for example: two (multiword) names are judged the same if one name is an initial subsequence of the other.

There are 31 such heuristic rules for matching organization names, 11 heuristics for person names, and three rules for location names.

Header Processing

Due to the use of capitalisation in article titles, the semantic representation produced by the parser is generally unreliable, with many capitalised words wrongly treated as proper names. For this reason processing of the header is delayed until after the body of the text, on the assumption that the true proper names in the header will also be mentioned, and more reliably detected, in the body. Proper names from the header which cannot be coreferred with anything in the body are then converted to normal predicate names and a further attempt made to find any coreferences.

Discourse Model

The processing of a text acts to populate the initial bare world model with the various instances and relations mentioned in the text. It is therefore converted into a world model specific to the particular text, i.e. a *discourse model*, containing the information necessary for the production of the results for all the MUC-6 tasks, and other potential applications.

Results Generation

The results for all four MUC-6 tasks are produced by scanning through the discourse model produced by the discourse interpretation stage. Most of the semantic classes and property types in the predefined ontology are motivated by distinctions required by the various tasks, and for the TE and ST tasks specific ontological properties have been introduced for each slot required. The results generation therefore only involves the retrieval from the discourse model of those instances which have all the required properties, and the correct formatting of the property values.

Named Entity Results

All instances of the ontological classes of organisation, location, person, etc. are retrieved from the discourse model. For each of these that has an `ne_tag` property, the value of which is a token range, an entry is added to an output file specifying the range and the required SGML markup type. The `ne_tag` property is introduced into the semantic representation of the text during the named entity stage of parsing. It is distinct from the `realisation` property, which also specifies a token range, because not all instances of the required classes should be output for the NE result, for example location names within organisation names. The `ne_tag` property is only assigned to those instances that should be in the output. At the discourse interpretation stage, ambiguous names are also assigned `ne_tag` properties, but these names will only be output in the NE result if they are subsequently classified as instances of one of the required classes.

The file containing the list of token ranges to be tagged is then used in conjunction with the original text and the output of the lexical preprocessing stage, to produce a new version of the text with the required SGML markup.

Coreference Results

The discourse model is searched for any instances of the object class which have more than one token range `realisation` property. This will only be the case where two distinct instances have been merged during discourse interpretation, resulting in a single instance with multiple realisations in the text. Instances of the event class also have `realisation` properties, but the MUC-6 task definition only requires coreferring noun phrases to be identified.

An output file is written specifying all the token ranges included in each coreference chain. As with the NE result, this file is used in combination with the lexical preprocessor output to produce a new version of the original text with the coreference SGML markup.

Template Element Results

All organisation and person instances are retrieved from the discourse model, and those with `name` properties are formatted as required and written out directly to a results file. Property values for the other slots, such as ORG_LOCALE, are searched for by examining other related properties, such as being situated *in* a location, and the values output if found.

Scenario Template Results

As for the template element result, this basically involves searching for instances which have values for the required properties and then writing them directly to an output file in the required format. In this case we require event instances of the type `succession_event`, which have values for the properties `succession_org`, `succession_post`, etc. Each of these events is associated with at least two IN_AND_OUT objects (one IN and one OUT), also represented as instances in the discourse model. Output is only generated if at least one of these objects has values for its required properties.

System Performance

The following table shows the scores for the four tasks. For the evaluation run the system processed 29 out of the 30 texts for the NE and CO tasks, and 98 out of 100 texts for the TE and ST tasks. This run produced the official scores, referred to here as 'incomplete'. Several texts were missed due to the omission of a trivial error trap that would have allowed the system to have continued at the next sentence on occurrence of a certain error, rather than at the next text. The system was re-run with the error trap included and the results kindly scored by the MUC-6 scoring team. The results of this run are referred to in the table as 'complete' scores, and are, of course, unofficial.

Official and unofficial scores for the four tasks:

Task	NE			CO		TE			ST		
	R	P	P&R	R	P	R	P	P&R	R	P	P&R
MUC6 official (incomplete)	84	94	89.06	0.51	0.71	66	74	69.80	37	73	48.96
MUC6 unofficial (complete)	89	93	91.01	0.54	0.70	68	74	70.80	37	73	48.96

While detailed evaluation of the contribution of system components to all tasks could not be undertaken before the conference, we have been able to partially analyse the behaviour of our NE subsystem. The table below illustrates the contribution of each of the system modules to the Named Entity task for the 30 (complete) NE texts. Setting 4 is the fully functional system setting.

No	Setting	Recall	Precision	P&R
1	List lookup only	37	74	49.61
2	1 + parsing	80	93	85.98
3	2 + name matching	88	93	90.83
4	3 + full discourse interpretation	89	93	91.01

Walkthrough

The following table shows the scores of the LaSIE system for the walk through text. Our official scores for this text were well below our average across all texts due to failure to classify correctly one proper name, which led in turn to missing two of the succession events. After the evaluation we enhanced the discourse interpreter of the system with one specific feature, as described in the NE section below, and re-ran it on the text. The scores for the enhanced system are also shown for comparison.

Official and enhanced scores for the walkthrough text:

Task	NE			CO		TE			ST		
	R	P	P&R	R	P	R	P	P&R	R	P	P&R
Walk through	79	94	85.91	0.69	0.86	54	63	57.89	13	58	21.88
Walk through (Enhanced)	94	95	94.41	0.70	0.86	63	68	65.82	50	84	62.65

Named Entity Task

The score for NE task is 79 recall, 94 precision and 85.91 for P&R. We missed one company name ('McCann-Erickson' and its abbreviation 'McCann') and one date expression. We recognized spuriously one company name ('Coca-Cola') and one person name. A company name was captured wrongly as person name ('J. Walter Thompson').

'Coca-Cola' in 'the prestigious Coca-Cola Classic account' was mistakenly recognized as company name since it is in the list of company names of the system and simply marked up as company at the list lookup stage.

'McCann-Erickson' was missed because the name itself does not have specific information to make it a company name and it remained an ambiguous proper name.

The system correctly recognized the two shortened forms of 'John J. Dooner Jr.', 'John Dooner' and 'Mr. Dooner' using the name matching algorithm in the coreference resolver.

We enhanced our discourse interpreter so that it recognizes an ambiguous proper name as a company name when it is preceded by a post name(s) and 'of', as in 'chairman and chief executive officer of *McCann-Erickson*'. With this enhancement, which in turn permitted the correct coreference resolution with 'McCann' as its abbreviation, the score, especially recall went up: 94 recall, 95 precision and 94.41 for P&R.

Coreference Task

Coreference scores for the walkthrough article are quite high compared to the overall system performance. The use of the enhanced system, which correctly recognises 'McCann-Erickson' as a company name, makes little difference to the walkthrough score. One additional coreference is correctly made between the names 'McCann-Erickson' and 'McCann', due to the recognition as a company which invokes the organisation specific name matching, as described earlier.

Obvious errors in the coreference include the omission of any appositions between person names and post titles in the first paragraph. This is due to the simple fact that post titles were not specified as being instances of the class `person` in the ontology, and so they could not corefer because of the lack of any path between the two classes.

Another error is the coreference of *it* at the beginning of the third paragraph, with *Yesterday* in the previous paragraph. Only paragraph initial pronouns are allowed to have coreferents outside the current paragraph, but in this case the use of the pronoun is not to anything specifically mentioned in the text. There is a more general problem of the non-referring use of *it*, as in "Mr. James says it is time...", which is not treated specially in the system here, and the pronoun is simply coreferred with the closest potential candidate.

Quotations also receive no specific treatment, leading to errors such as the *I* in "I Can't Believe It's Not Butter" being coreferred with the last person mentioned in the text, regardless of the fact that the quoted text is not attributed to anyone.

Most of the heuristic rules for the coreference task, implemented via the properties of classes in the ontology, were produced from training on the MUC-6 dry-run articles, and few were subsequently modified. The restriction to only attempt coreferences within the two most recent paragraphs (apart from instances with names), as described earlier, was introduced at a later stage to reduce the processing time of the discourse interpretation phase, resulting in a predictable slight loss of recall with a corresponding increase in precision.

Template Element Task

The score for TE task is 54 recall, 63 precision and 57.89 for P&R. The score reflects, to large extent,the successes and mistakes which are made at the NE task.

TE task specific organization descriptors were not captured well in the system and this caused the instantiation of two spurious organizations. As for locale and country slots, when a location name appears near an organization name, it will be associated with the organization. However, the two location names are missed for the text.

As long as the names are correctly recognized, their aliases are all correctly recognized. 'Coke' for 'Coca-Cola' is matched using a list of difficult alias names and 'Mr. Dooner' is matched with its full name, 'John J. Dooner Jr.' accurately.

One interesting name in the text is 'J. Walter Thompson'. Clearly this is a person name however, it appears in the text as '... was hired from WPP Group's *J. Walter Thompson* last September' and here it names a company. Our system does recognize it as person name but it does not change it to a company name.

Scenario Template Task

The system's performance in this task for the walkthrough article is poor, producing a P&R score of only 21.88 compared to the overall ST P&R of 49.27. However, with the enhanced version of the system P&R is raised to 62.65 for this article.

The original poor result is due to the failure to identify the names 'McCann-Erickson' and 'J. Walter Thompson' as company names. The use of the verb *hire* in the following piece of text

> *Peter Kim was hired from WPP Group's J. Walter Thompson last September as vice chairman, chief strategy officer, world-wide*

triggers the creation of a `succession_event` in the discourse model. This creates two `in_and_out` objects, with the `IN` object associated with 'Peter Kim', because the verb *hire* is classified as an `IN` verb in the ontology. The properties associated with this ontological class specify that the phrase "as vice chairman" following the verb indicates the `succession_post` of this event. The properties of the `hire` event also specify that any "by" phrase would indicate the `succession_org`, but in this case no such phrase is found. Properties of the more general `succession_event` class are then used to introduce a dummy organisation into the discourse model which the system then attempts to corefer in the same way as any other instance, in this case selecting the most recently mentioned company as the coreferent, and therefore the `succession_org`, which, due to the omission of McCann, is in fact 'Coca-Cola'.

`succession_events` are also created in the discourse model for each of the verbs *step_down*, *retire* and *succeed* in the second paragraph of the text. However, because McCann is not known to be a company, no `succession_org` is found for these events, and no output is produced, `succession_org` being one of the compulsory slots. In the enhanced version of the system, where McCann is recognised as a company, the `succession_event` created by *retire* does have values for all its properties, and it is output correctly. An attempt is made to merge this `succession_event` with the one created by the use of *succeed*, with the intention of providing values for both the `IN` and `OUT` objects of the *retire* event, but the merge fails in this particular case.

This approach to the scenario task involved the manual classification of 27 `succession_event` verbs, most of which had specific patterns associated with them, via the property mechanism, for establishing the person and organisation involved. A small set of general defaults could also be inherited by the events if the specific patterns did not apply. The scenario patterns are heavily reliant on the accuracy of the named entity recognition, although the ability is included to convert an ambiguous name in a specific verb pattern into a person name, and this could be extended further. This reliance results in the reasonable precision score of the system for this task, but it is at the expense of recall.

Scenario Template Summarisation

An additional result produced by the system is a natural language summary of the text driven by the information contained in the scenario template. For each succession event found a simple sentence is produced, possibly referring to information in the discourse model which is not contained in the scenario template itself, for example:

```
<SUMMARY-9402240133> :=

Robert L. James steps down as chief executive officer of McCann-Erickson.

Robert L. James will retire as chairman of McCann-Erickson.

McCann-Erickson hired Peter Kim as vice chairman.
```

Observations

Like most MUC systems, LaSIE was improving rapidly in performance up to and, as our enhancements to the system for the walkthrough article show, after the final evaluation. So it is difficult to judge the limitations or still-to-be-realised potential of the underlying techniques.

What is clearly needed is yet more experimentation to determine just where the critical areas in performance are. In particular we need to assess the adequacy of the grammar and of our 'best parse' selection algorithm: this could be done by doing *parseval*-style evaluations against the Penn Treebank. We also need to attempt to evaluate and improve the algorithms for automatically extending the ontology.

In our view MUC-6 has provided an extremely valuable increase in our understanding of information extraction systems and the inter-relation of their components. We expect to be learning from our results for some time to come.

Acknowledgements

This research has been made possible by the grants of the U.K. Department of Trade and Industry (Grant Ref. YAE/8/5/1002) and the Engineering and Physical Science Research Council (Grant # GR/K25267). We would like to thank BBN who have allowed us to use their POST program.

References

[Bri94] E. Brill. Some advances in transformation-based part of speech tagging. In *Proceedings of AAAI*, 1994.

[CGJ+93] J. Cowie, L. Guthrie, W. Jin, R. Wang, T. Wakao, J. Pustejovsky, and S. Waterman. Description of the *Diderot* system as used for MUC-5. In *Proceedings of the Fifth Message Undersanding Conference (MUC-5).* ARPA, Morgan Kaufmann, 1993.

[Gai95a] R. Gaizauskas. Investigations into the grammar underlying the Penn Treebank II. Research Memorandum CS-95-25, Department of Computer Science, Univeristy of Sheffield, 1995.

[Gai95b] R. Gaizauskas. XI: A knowledge representation language based on cross-classification and inheritance. Research Memorandum CS-95-24, Department of Computer Science, Univeristy of Sheffield, 1995.

[GCE93] R. Gaizauskas, L.J. Cahill, and R. Evans. Description of the sussex system used for MUC-5. In *Proceedings of the Fifth Message Undersanding Conference (MUC-5).* ARPA, Morgan Kaufmann, 1993.

[Hor80] A.S. Hornby, editor. *Oxford Advanced Learner's Dictionary of Current English.* Oxford Univeristy Press, 1980.

[Mil90] G. Miller. Wordnet: An on-line lexical database. *International Journal of Lexicography*, 3(4), 1990.

[MKM+95] M. Marcus, G. Kim, M.A. Marcinkiewicz, R. MacIntyre, A. Bies, M. Ferguson, K.Katz, and B. Schasberger. The Penn Treebank: Annotating predicate argument structure. Distributed on The Penn Treebank Release 2 CD-ROM by the Linguisitc Data Consortium, 1995.

[MSM93] M.P. Marcus, B. Santorini, and M.A. Marcinkiewicz. Building a large annotated corpus of english: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993.

[WMS+93] R. Weischedel, M. Meteer, R. Schwartz, L. Ramshaw, and J. Palmucci. Coping with ambiguity and unknown words through probabilistic models. *Computational Linguistics (Special Issue on Using Large Corpora: II)*, 19:359–382, 1993.