UNIVERSITY OF SHEFFIELD: DESCRIPTION OF THE LaSIE-II SYSTEM AS USED FOR MUC-7

K. Humphreys, R. Gaizauskas, S. Azzam, C. Huyck, B. Mitchell, H. Cunningham, Y. Wilks [1]
Department of Comp uter Science
University of Sheffield
Regent Court, Portobello Road
Sheffield S1 4DP UK
{kwh,robertg,saliha,huyck,brianm,hamish,yorick}@dcs.shef.ac.uk

INTRODUCTION

The University of Sheffield NLP group took part in MUC-7 using the LaSIE-II system, an evolution of the LaSIE (Large Scale Information Extraction) system first created for participation in MUC-6 [9] and part of a larger research effort into information extraction underway in our group. LaSIE-II was used to carry out all five of the MUC-7 tasks and was, in fact, the only system to take part in all of the MUC-7 tasks.

While LaSIE-II is significantly different from the earlier version (differences are detailed below) there are no radical changes in the basic philosophy of the approach. This could be described as seeking a pragmatic middle way in the shallow vs deep analysis debate which has characterised the last several MUCs. That is, while aware that information extraction tasks may not require full text understanding, and hence that systems should be optimised to make use of shallow techniques where appropriate, we have not wanted to preclude the application of arbitrarily sophisticated linguistic analysis techniques where these may prove useful. The result is an eclectic mixture of techniques including finite state recognition of domain-specific lexical patterns, partial parsing using a restricted context-free grammar, simplified semantic representation of each sentence in the text and a formal representation of the whole discourse from which all of the IE task results and the coreference task results are derived. From our perspective, LaSIE-II should not be viewed as the expression of a theory about how to do IE, but as a laboratory in which ongoing experiments with different component NL processing techniques, and most importantly, *their interaction* are being carried out. Seen this way, one of the most important developments in LaSIE-II is its modularised architecture and integration into the GATE platform (see below) which has enabled us to gain much deeper insights into strengths and weaknesses of components of the system and the ways in which these interact.

OVERVIEW

LaSIE-II is a highly modularised system, made up of 9 TIPSTER-compliant modules, pictured in Figure 1 as executed interactively through the GATE Graphical Interface. The system is essentially a pipeline of modules each of which processes the entire text before the next is invoked. The following is a brief description of each of the component modules in the system:

Tokenizer Identifies token boundaries (as byte offsets into the text) and text section boundaries (text header, text body and any zones to be excluded from processing).

[1] Thanks for additional contributions from Sandy Robertson, Andrea Setzer, George Demetriou, Malcolm Crawford ({sandyr, andrea, demetri, malc}@dcs.shef.ac.uk) and Mette Nelson (mln.id@cbs.dk) from the Copenhagen Business School.

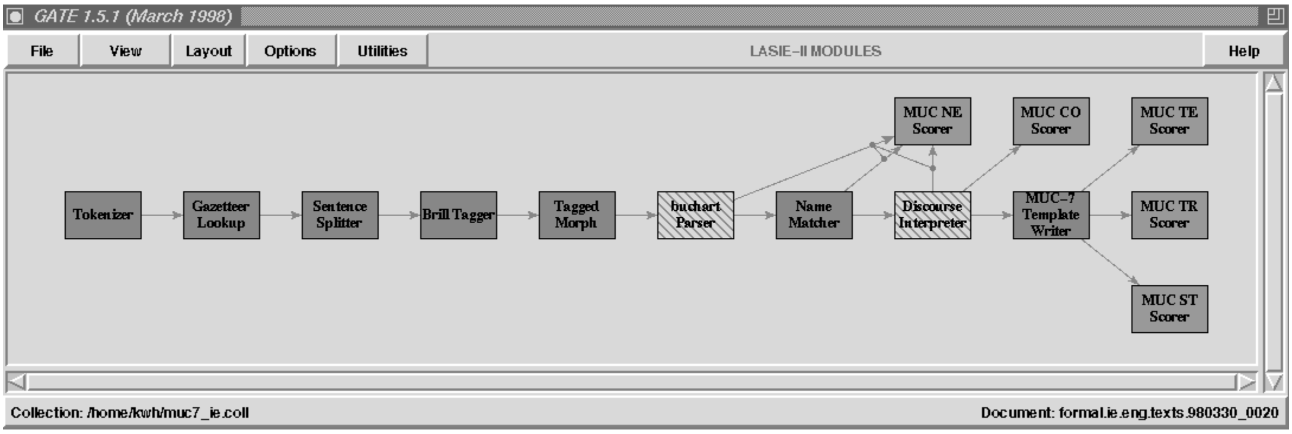

Figure 1: LaSIE-II Architecture

Gazetteer Lookup Looks for single and multi-word matches in multiple domain specific full name (locations, organisations, etc.) and keyword (company designators, person first names, etc.) lists, and tags matching phrases with appropriate name categories.

Sentence Splitter Identifies sentence boundaries in the text body.

Brill Tagger [4] Assigns one of the 48 Penn TreeBank part-of-speech tags to each token in the text.

Tagged Morph Simple morphological analysis to identify the root form and inflectional suffix for tokens which have been tagged as noun, verb, or adjective.

buchart Parser Does two pass bottom-up chart parsing, pass one with a special named entity grammar, and pass two with a general phrasal grammar. A 'best parse' is then selected, which may be only a partial parse, and a predicate-argument representation, or quasi-logical form (QLF), of each sentence is constructed compositionally.

Name Matcher Matches variants of named entities across the text.

Discourse Interpreter Adds the QLF representation to a semantic net, which encodes the system's domain model as a hierarchy of concepts. Additional information presupposed by the input is also added to the model, then coreference resolution is performed between new and old instances, and finally information consequent upon the input is added, producing an updated discourse model.

Template Writer Writes out the ST, TR, and TE results by traversing the discourse model and extracting the required information.

NE and CO results are generated following the Discourse Interpreter by a generic SGML dump utility.

LaSIE-II CHANGES

Rather than duplicating much of the description of the LaSIE system as used for MUC-6 [9], the following sections describe the major changes between LaSIE, referred to in the following as LaSIE-I, and LaSIE-II.

GATE/TIPSTER Architecture

The LaSIE-II system was developed using GATE, a General Architecture for Text Engineering [5], Sheffield's implementation of the TIPSTER architecture specification [11]. GATE manages all the information about the texts that is produced by each module, and provides graphical tools for visualising that information, selecting control flow through different module combinations and running the IE system over sets of texts. A major strength of the architecture is that it encourages reuse by insulating the various modules from each other by means of a common data management substrate — a "document manager" in TIPSTER terminology. It also enables reuse of visualisation code: modules producing similar sorts of information (e.g. a PoS tagger and a named entity parser) share the same graphical viewing tool. GATE provides a convenient GUI-based environment within which to develop diverse modules, unconstrained by implementation language (LaSIE-II is made up of C, Perl and Prolog modules), with the architecture taking care of the common engineering tasks (e.g. data storage) that are uninteresting from a language processing point-of-view. Lastly, GATE provides a command-line interface for batch processing, enabling us to run a nightly build, run, score, and report process as we developed the LaSIE-II system.

GATE is now in a one-and-a-half release (1.5) that adds Java support, SGML I/O, a manual annotation tool, an annotation comparison tool and improved support for managing collections of documents. This release is a half-way house between version 1, which was C++-based, and version 2, which will be Java-based. Java modules that run under version 1.5 will run unchanged under version 2, while still accessing all the modules and facilities available under version 1. SGML facilities are much improved, with input via the University of Edinburgh's LT NSL toolkit [15] that uses the Sp parser. The manual annotation tool allows hand-coding of annotations to use as training or test data, and the comparison tool provides a straightforward way to score one annotation set against another. In cases where a dedicated scorer is available, like MUC or Parseval [12], this can be integrated as a module in GATE, as shown in Figure 1 with a scorer module for each MUC task. The output of the MUC scorers can also be read into the GATE database, allowing keys and errors to be displayed using the existing viewers, as shown for the CO task scorer in Figure 2. Wrapper code for these and other modules is available from our `ftp` site (see `www.dcs.shef.ac.uk/research/groups/nlp/gate/` for more details, and to download GATE, which is freely available for research purposes, and comes bundled with a version of our MUC-6 extraction system).

Lexical Preprocessing

With the exception of the Gazetteer Lookup stage, modules up to the buchart Parser are relatively unchanged from LaSIE-I. Minor changes were required to make use of the structure of the MUC-7 NYT texts, and to classify SGML and other special symbols.

The most noticeable change to the Gazetteer Lookup module is its change in position – from immediately before the Parser in LaSIE-I to immediately after the Tokenizer in LaSIE-II. This change was made mainly to improve the accuracy of the Sentence Splitter module, which could previously propose incorrect sentence boundaries within known NEs, for example "3 p.m. EST", "St. Louis", etc. The Splitter has been modified slightly to treat gazetteer matches as units in the same way as tokens.

The move involved decoupling the Lookup stage from the PoS Tagger. Originally only tokens with particular tags (nouns, adjectives, determiners, conjunctions, numerals, symbols) were matched against the gazetteer lists, but this restriction has been removed. The Lookup stage now attempts to match all tokens, and therefore no longer suffers from tagging errors. However, this does introduce a few spurious matches, particularly with capitalised words in sentence initial position, for example the (Swedish) person first names "Are" and "My". An additional filtering stage in the Tagger module handles these cases, removing any gazetteer matches for sentence initial tokens not tagged as nouns or adjectives. This filtering stage also attempts to correct some of the tagger's frequent mistagging of capitalised common nouns as proper nouns in document headers, by reference to a list of common English nouns (also used in the Splitter module).

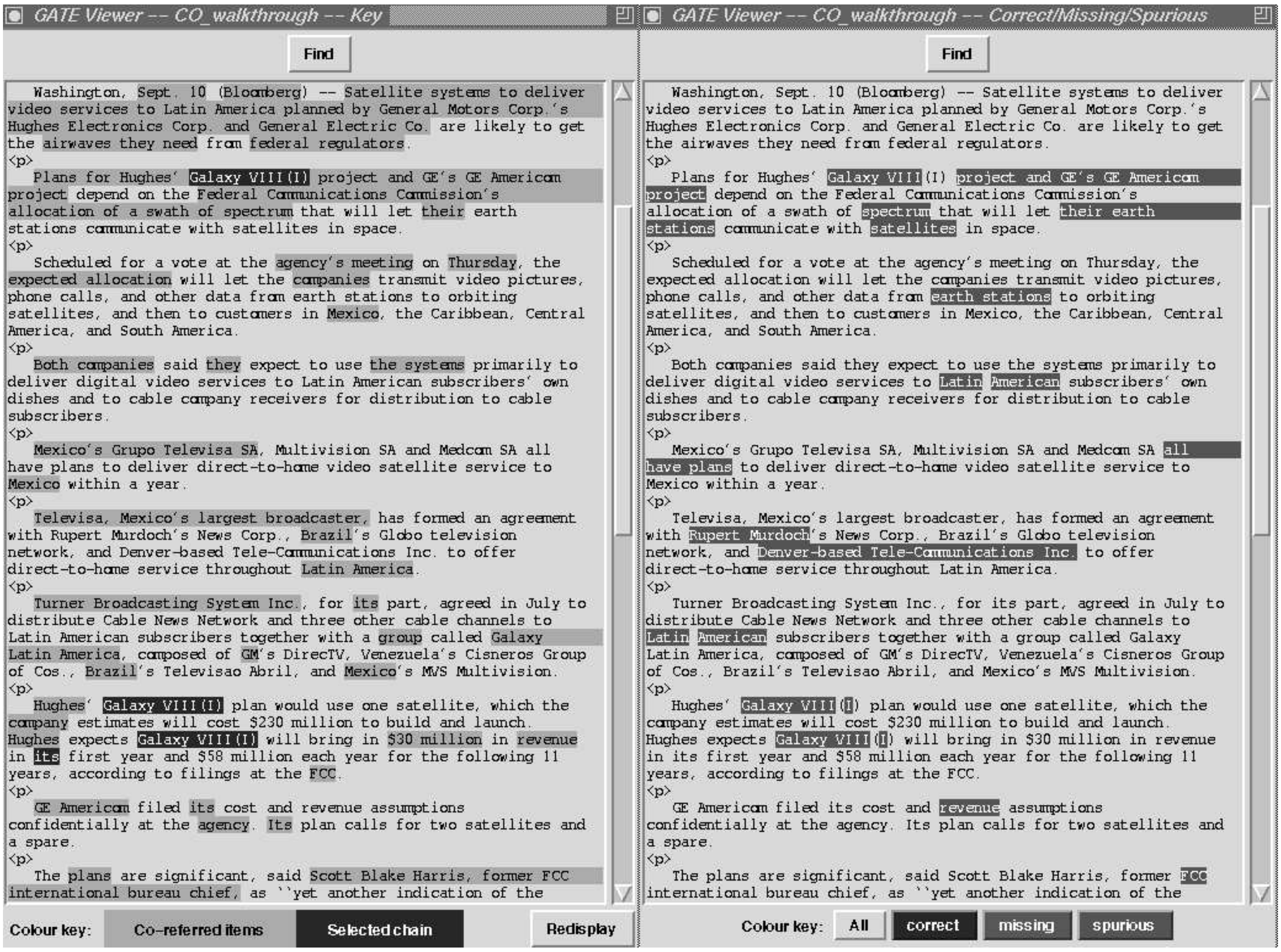

Figure 2: MUC CO Scorer output viewers, showing spurious coreferences during debugging

The Lookup stage has also been modified to allow much simpler integration of new lists of names, each defining a distinct semantic category. A top-level configuration file specifies a set of plain text lists and type and subtype values (e.g. organization:company) to be assigned to matches in each list. The module can now be switched between domains simply by specifying alternative configuration files. For MUC-7 we used 55 lists comprising 23,000 entries in total.

A further improvement to the Lookup module is a reduction in its case sensitivity. While initial experiments with complete case insensitivity in matching against the lists produced too many spurious matches, some reduction in sensitivity proved useful. In particular, sequences of all-uppercase tokens in the input are now matched in the lists in both their original form and also with each token converted to a form where only the initial character is uppercase. This significantly improves matches in the NYT headers.

Parsing

As in LaSIE-I, parsing is still carried out in two stages, each stage using the same parsing mechanism but a different grammar – first a specialist NE grammar, then a general phrasal grammar. The parser itself is largely unchanged from MUC-6 – a bottom-up chart parser written in Prolog which processes context-free grammar rules with associated feature structures expressed as Prolog terms. A complete chart is generated from which a single parse, quite possibly partial (i.e., a gapped sequence of phrasal subtrees), is selected using a 'best parse selection' heuristic when parsing ceases. Semantic interpretations in the form of predicate-argument representations are built up compositionally from phrasal constituents during parsing and the semantic interpretation of the final selected analysis becomes the output of the parser and gets passed on to the discourse interpreter.

The main changes introduced for MUC-7 are a significantly enhanced grammar development environment, and a completely rewritten and extended grammar which now reflects a substantially different philosophy.

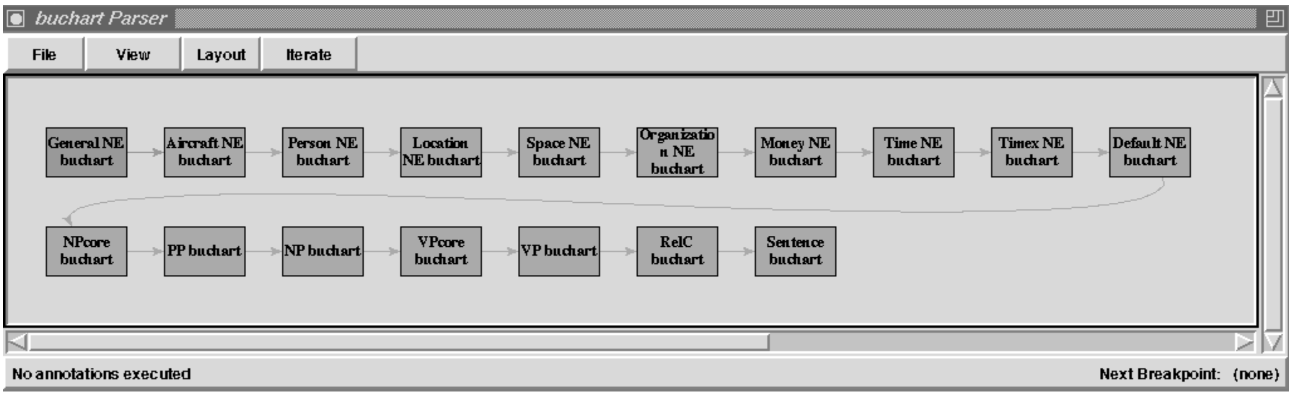

Figure 3: buchart Parser submodules

Grammar Development The modular facilities of GATE have been exploited to further compartmentalise the grammars into a total of 17 subgrammars which during development can be individually executed through the GATE graphical interface (see Figure 3). After each grammar is run, its results may be viewed using a tree viewer, and then, if changes are required, the grammar may be edited and rerun without leaving GATE for any recompilation process.

The first ten grammars shown in Figure 3 comprise the NE grammar and when the system is run in production mode the rules from these ten grammars are compiled into a single grammar for use in the first pass of the parser. The net effect is the same as running them separately since the ordering of the rules in the compiled single grammar is the same as in the cascaded development version and the best parse selection heuristic will, other things being equal, select the last of co-spanning analyses. The same holds for the next seven subgrammars which form the phrasal grammar and are compiled into a single grammar for the second pass of the parser in production mode.

Division of the grammar into smaller specialist chunks together with the new graphical tools had the expected benefits of allowing more rapid development and verification of subgrammars, and supported concurrent grammar development by different persons working on different subgrammars.

The Grammars The ten NE grammars consist of approximately 400 hand-coded rules that make use of part of speech tags, semantic tags added in the gazetteer lookup stage, and if necessary lexical items themselves. While significantly rewritten since MUC-6 the basic philosophy here is the same (see [17] for details): patterns are detected in the texts and manually added to the grammar. The enhancements to the grammar development cycle described above have eased and speeded this process, but otherwise there is little change.

The phrasal grammars have been completely rewritten and compartmentalised, but there has also been a significant change in the grammar acquisition process. For MUC-6 the grammar was obtained by extracting the context-free grammar rules implicit in the bracketing of the Penn Tree Bank [14] and selecting a subset of them by thresholding on frequency [6]. Features to enable semantic interpretation were then added by hand to the extracted syntactic rules.

While this technique allowed us to obtain a grammar quickly, the resulting analyses were poor, and the effort of manually annotating the rules with features for semantic interpretation substantially reduced the benefits of the 'grab-and-run' approach to grammar acquisition. Given that so much handcrafting was going into the grammar, it seemed that we might as well get the benefit of more carefully handcrafting the syntactic aspects of the rules too. So, we have effectively rewritten the grammar from the ground up, using a combination of general principles [16] and iterative refinement using the MUC-7 training data. The result is a phrasal grammar of about 150 rules.

The best parse selection heuristic chooses a single (possibly partial) analysis based on selecting a shortest sequence of maximally spanning non-overlapping edges which are of a semantically interpretable category (NP, VP, PP, S, and RelC). Where there are several equivalent alternatives, the last one generated is selected. This approach eliminates any ambiguity detected in parsing and ensures that a single analysis is passed on to the discourse interpreter. However, since our grammar does not rely on lexical semantic or syntactic constraints to any great extent (e.g. we do not use lexical selectional restrictions or subcategorisation information in parsing) it is very weak. As a consequence, we adopt a very conservative approach with regards to phenomena such as attachment of complements, prepositional phrases and relative clauses, and also apposition and co-ordination. We followed a philosophy of only adding those rules which were (almost) certain never to generate errors in analysis – i.e. a high precision, possibly low recall, approach to grammatical analysis. In theory, when grammatical relations such as complements (e.g. subjects and objects) are missed, they are meant to be added during discourse interpretation, where lexical-semantic information is available, and to some extent this happens (see below), though we have not developed this part of the system as much as we would like. Exploring the boundaries between syntactic analysis, lexical-semantic analysis, and discourse analysis is very much part of ongoing work in LaSIE-II.

The net effect using a cascade of grammars, each of which aims to identify a 'chunk' of a particular category and is conservative with respect to attachment, is something very like the finite state models that have been advocated by other MUC participants over the past few years [13, 10], as well as others in the language engineering community [1]. In fact, we believe our grammars are now regular and that our chart parser could be replaced by a finite state parser, with substantial increase in speed. We have not done so to date due to restricted development resources and because the current grammar development/parsing environment is quite habitable.

Discourse Interpretation

Apart from some gazetteer lists, and the corresponding grammar rules, all domain specific knowledge in the system is concentrated in the Domain Model of the Discourse Interpreter. As in LaSIE-I, this model is expressed using a semantic net whose nodes represent 'concepts' (classes or instances), with associated attribute-value structures recording properties and relations of the concept, and whose arcs model a concept hierarchy and support property inheritance (see [7] for further details).

The initial domain model for the MUC-7 ST task was constructed directly from the template definition. A `launch_event` concept node was added, together with `vehicle` and `payload` nodes, each with a subhierarchy listing the possible `vehicle_` and `payload_type`s specified in the template definition:

```
entity(X) ==> object(X) v event(X) v property(X).
object(X) ==> artifact(X) v ...
event(X) ==> launch_event(X) v ...

artifact(X) ==> vehicle(X) v
                payload(X).
vehicle(X) ==>  spacecraft(X) v
                aircraft(X) v
                ground_vehicle(X) v
                water_vehicle(X).
spacecraft(X) ==> shuttle(X) v
                  rocket(X).

payload(X) ==> satellite(X) v
               missile(X) v
               space_probe(X) v
               material(X) v
               personnel(X).
```

Property types were also added for each template slot, `launch_date`, `vehicle_owner`, etc., and the Template Writer module was modified to read off and write out instances of the required types with the required properties.

Consequence properties (see below) were then added to hypothesise instances for each slot of a template entity, given the appropriate textual trigger. For example, an instance of a `launch_event` in a text causes the hypothesis of a vehicle, a payload, a date and a launch site related to the event; an instance of a `vehicle` causes the hypothesis of an owner and a manufacturer, etc. The Discourse Interpreter's general coreference mechanism is then used to attempt to resolve hypothesised instances with instances mentioned in the text. Running the system in this state, with absolutely no domain specific restrictions on the resolution of hypotheses, gave an overall performance of 27.66 P&R on the ST training data. This result was achieved using only the template definition and no training data to customise the system, and required only a few hours work. This baseline customisation of the system to a new IE task is largely mechanical, and we intend to investigate the extent to which initial concept hierarchies and associated properties can be produced automatically from a template definition.

A small set of predefined, or static, instances were also added to the Domain Model, encoding certain world knowledge necessary to complete particular ST task slots. For example, *NASA* was predefined to allow its use as a default value for the `payload_owner` slot for American astronauts, as required by the ST task specifications. Similarly, common spacecraft launch sites were predefined, restricting the selection of `launch_site` values. Much greater use of this facility could be made to encode other relevant world knowledge.

During processing, the instances and properties from the semantic representation of a text (QLF) produced by the Parser are added to the Domain Model. The semantic representation of each sentence is

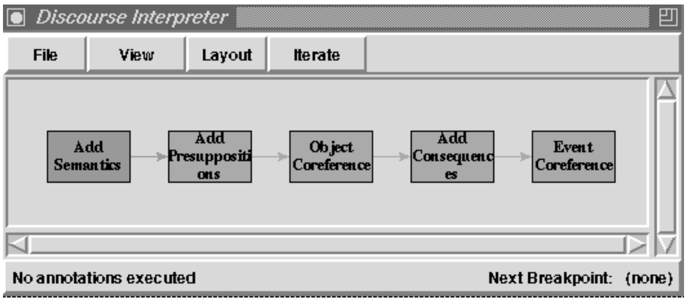

Figure 4: Discourse Interpreter submodules

processed in the following stages, illustrated by the submodules in the GATE interface shown in Figure 4, gradually specialising the Domain Model to become a Discourse Model, the final version of which is passed to the Template Writer.

Add Semantics Instances from the QLF representation of a sentence are added below their parent classes in the concept hierarchy. New concept nodes are created dynamically for classes not defined in advance, and are added directly below the `object` node for instances introduced by nouns, and below the `event` node for instances introduced by verbs. Properties in the input semantics are added to the attribute-value structures associated with the instances to which they relate.

A new mechanism introduced in LaSIE-II is a word root to concept node mapping, used to establish the parent nodes of new instances. Previously, concept nodes in the ontology were labelled with English word roots onto which the QLF semantic representation was mapped directly, forcing subclass hierarchies to be constructed even for synonymous terms. The introduction of a word-to-concept table, or dictionary, provides a many-to-one mapping onto the concept nodes in the ontology, allowing synonym sets to be represented straightforwardly in the table.

The word-to-concept mapping also provides the ability to process QLF from non-English languages with the same Discourse Interpreter and Domain Model. An experimental Multilingual LaSIE (M-LaSIE) system has been constructed to process French, Spanish and English texts, producing templates or natural language summaries in each language, using the word-to-concept table for output as well as input. Further details on this system can be found in [8].

Add Presuppositions Each instance and property added from the current sentence semantics attempts to inherit any presuppositions, predefined in the domain model, from its parent classes. Presupposition properties of concept nodes are used to perform the following functions in the discourse model:

1. *Additional Inferences* Add inferred information to the current discourse model, in addition to the input semantics. For example, a presupposition of the property `name` (proper name) is that any instance with this property must be an instance of the `object` class, and also have a `number` property with the value `singular`. If not already known, this information will be added to the discourse model.

   Some inferences will be highly domain, or even template, specific, providing particular slot values based on patterns of semantic relations in the input.

2. *Entity Hypothesis* Expected, or implicit, instances, can be hypothesised to be resolved by the coreference mechanism. Nominalisations of verbs can be identified by presuppositions and lead to the hypothesis of the corresponding event, for example a hypothesised `launch_event` from an instance of a `launch` object. Such events, if they acquire the necessary properties, will be written out as ST

template entries. Similarly, instances of indirectly related scenario specific objects, such as `mission`, can also give rise to the hypothesis of a `launch_event`.

3. *Word Sense Disambiguation* Identify whether an instance of a particular class in the input is the same as a known class in the ontology. For example, the ontology contains `fall` as a subclass of `date`, but an instance of `fall` initially added here may be removed by a presupposition that identifies this instance as, say, referring to a fall in share prices rather than a date. Scenario specific senses can also be caught in this way, for example only `fire_event` instances related to missiles are retained as potential `launch_event`s.

4. *Role Classification* The ontology contains a hierarchy of person roles, including domain specific roles such as astronaut, pilot, etc. A presupposition acts to reclassify instances of these nodes from the text as instances of the `person` node, with a property indicating the role. This avoids the previous requirement to specify person roles as subclasses of person to force semantic compatibility for coreference. Subhierarchies of roles can now be defined, for example job roles and family roles, with cross classification of instances permitted. The roles used for the ST task were obtained by identifying the intersection of terms in the ST training data with entries in an electronic dictionary with an animate feature.

5. *Partial Parse Extension* Missing or unattached properties will cause the hypothesis, and attempted resolution, of required instances. A presupposition of the `event` node specifies that all instances below it should have an `lsubj` (logical subject) property. This presupposition will cause the hypothesis of a new instance for each `event` instance that the parser has failed to attach a subject relation to. The general coreference mechanism is then used to attempt to resolve the hypothesis, applying various restrictions, also expressed as properties in the domain model, such as requiring subjects to be before an active verb in the same sentence. More specific event types can presuppose additional restrictions, for example restricting a hypothesised subject of a `crash_event` to be of type `vehicle`. This effectively specifies semantic roles, or subcategorisation patterns, for particular event types. However we currently specify very few such restrictions.

   The same mechanism can also be used to attempt prepositional attachment, where the parser has left a phrase unattached. Phrases can be classified as temporal, locational, etc., using the semantic classes of their heads, and then semantic role information used to identify potential attachment sites, as for verb arguments. Again, however, we make little use of this facility in the current domain model.

Object Coreference The general coreference mechanism takes a set of instances newly added to the discourse model, and compares each one with the set of instances already in the discourse model. For object coreference, proper names, pronouns, and common nouns are handled separately, first attempting intra-sentential coreference for each set, and then inter-sentential coreference. Each new-old pair of instances, if at all compatible, has a similarity score calculated for it, based on the distance between the instances' parent classes in the concept hierarchy, and the number of shared properties. The highest scoring pair, for each new instance, is merged in the discourse model, deleting the instance with the least specific class in the ontology, and combining the properties of both instances. This mechanism is basically unchanged from the LaSIE-I MUC-6 system, but with the addition of several new features:

Coreference

1. *Long Distance Coreference* In LaSIE-I antecedents for pronouns and bare nouns were sought only in the current and immediately preceding paragraphs, and no attempt was made to find an antecedent in earlier paragraphs even if the anaphor almost certainly required one, as in the case of pronouns. This has been extended to search successively earlier paragraphs until an antecedent can be found. On the

30 dry run texts, this extension gave a 2% increase in recall for pronouns, and a 7% increase in recall for bare nouns, with no significant change in precision.

2. *Copular Constructions* Constructions of the type *NP1 be NP2* where *NP1* should corefer with *NP2* ('Predicate Nominals' in the CO task definition), e.g. *The F14 "Tomcat" is the Navy's first-line fighter aircraft*, were not dealt with in LaSIE-I due to lack of development time, but they are now considered by the coreference algorithm.

   This necessitated reviewing all the coreference rules, to add exceptions for copular constructions. For example, in general an indefinite noun phrase such as *a president* cannot have an antecedent, but this needed to be relaxed so as not to apply to copulars.

   Another important aspect of copular constructions is that they provide information to allow 'unknown' words, i.e. words whose semantic class is not in the ontology, to be classified during processing. This is possible when an instance of an unknown class is coreferred in a copular construction with an instance of a known class. For example, in *Bill is a president*, if `president` is not known as a concept in the ontology and *Bill* is recognised as an instance of the known `person` node, then a `president` node can be automatically added below `person`. Subsequent coreferences can then be more accurate by preferring or preventing coreference with instances of the newly added class, for example, to prevent subsequent occurrences of *it* from being resolved with instances of `president`.

3. *Catapora* LaSIE-II handles two specific cases of pronouns occurring before their antecedents. Firstly, pronouns within quotations, such as:

   "I caught Reggie when he was much younger counting his dad's trophies," McNair said.

   where *I* refers ahead to the speaker, *McNair*. However, the coreference is currently only possible if a complete sentence is successfully parsed within the quotation. Secondly, pronouns within copular constructions can also refer ahead, as in:

   This is a mystery.

4. *Coordinated NPs* A change in the MUC-7 CO task specification was the introduction of certain conjoined NPs as markables. In the LaSIE-II coreference algorithm, a new instance representing the set of any coordinated instances is created in the discourse model, which can act as a potential antecedent. The new instance will have a `plural` attribute and the semantic class will be the lowest common parent class of the coordinated instances. For example, in *Bruce and his boys* three instances: `e1`, *Bruce*, `e2`, *his boys*, and the set instance `e3`, *Bruce and his boys*, of type `person` will be represented in the discourse model.

   However, coreferences may fail or spurious coreferences be generated if the parser fails to correctly recognise the coordinated phrases on which the set instance identification relies.

5. *Header Coreference* In LaSIE-I only proper noun coreferences was attempted in text headers, but in LaSIE-II both pronoun and bare noun coreferences are also attempted. However, coreference rules that apply for normal sentences may fail for header sentences because of incomplete syntactic information caused by the telegraphic style and omission of determiners common in headers. Relaxing some of the coreference rules for headers gave a noticeable improvement in recall.

6. *Generic Nouns* A new subclass of bare nouns was introduced, with its own set of specific resolution rules. These 'generic' nouns occur as NP heads with no modifier or other relation which could allow additional syntactic or semantic information about them to be inferred.

The coreference mechanism still has a number of limitations. One of the most common pronoun errors is related to first and second person pronouns (*you, we, I*), but there is currently no special treatment for this class. Also, these pronouns typically occur within quotations, the treatment of which is still very limited.

We also do not attempt to handle type coercion and metonymy, and so fail to resolve pronouns like *they* in cases such as:

> *The Navy informs me that they have been unable to find a common thread to these accidents.*

We also need a more robust proper name matching module, since the coreference mechanism is heavily dependent on this for proper name coreference. It currently fails to match the organisation *National Transportation Safety Board* with *Safety Board*, while it does match *Latin America* with *Latin* — errors which could be easily corrected.

Add Consequences The use of consequence rules parallels that of presuppositions, but since they apply after object coreference, they may refer to information outside that of the current sentence. If an instance in the current input is resolved with an instance elsewhere in the text it will acquire additional properties, potentially allowing more accurate inferences.

The majority of template slot fills are proposed through consequence properties, and thus most of the domain specific rules are applied at this stage. Consequences can hypothesise unknown instances in the same way as presuppositions, but these hypotheses are not restricted to being resolved during the processing of the current sentence. Unresolved hypotheses from consequence rules are retained and resolution retried after each subsequent sentence. As a special case, unresolved hypotheses may be removed by the introduction of new hypotheses of the same type but from a different source. Currently, this removal only takes place for hypotheses from consequences of `launch_events`, to reflect an assumption that instances related to a first launch will not be introduced in a text following the description of a second launch. If, however, two mentions of an event are subsequently coreferred, all properties, including unresolved hypotheses, will be merged anyway.

Event Coreference The final stage in the processing of each sentence is to attempt to merge `launch_events` introduced in the current sentence with any others in the discourse model. This includes both hypothesised and explicitly mentioned events. Initially the criterion for merging two events was that all related entities (vehicle, payload, site, date) of each event be equivalent. However, this proved to be too strict, generating many spurious template fills due to failed merges, and the criterion was gradually relaxed until it was required only that the payload of two events be the same. This makes a strong assumption that a payload will never be related to more than one launch event. Other related entities are not compared, and only the entities related to the earlier of two events are retained in a successful merge.

RESULTS

The following table summarises LaSIE-II system performance on all five tasks. After the NE evaluation a few modifications were made to the initial system (system 'A') as a result of examining the IE training data which had been the NE formal run data. In particular, a name list for 'astronomical bodies', none of which were identified during the NE evaluation run, was added when we realised we should have been marking these as locations; we also added a list of rocket names to help with identifying artifacts for the TE task. Thus, the 'A' system results for NE are the official results; the 'B' system (the system run in the subsequent CO and IE evaluation) results are official for the other tasks. We unofficially rescored the 'B' system against the (no longer blind) NE evaluation data to see what effect the changes had.

Task	System	Recall	Precision	P&R
NE	A	83	89	85.83
	B	87	94	90.41
CO	B	56.1	68.8	61.8
TE	B	75	80	77.17
TR	B	41	82	54.70
ST	B	47	42	44.04

WALKTHROUGH TEXTS

Named Entity

Task	System	Recall	Precision	P&R
NE Walkthrough	A	77	86	81.40
NE Overall	A	83	89	85.83
NE Walkthrough	B	88	88	88.41
NE Overall	B	87	94	90.41

Performance on the NE walkthrough text was slightly below the level across the whole NE formal run test set. The system missed the organizations *Intelsat* and *Globo*, the locations *Latin America* and *Xichang*, and the person *Llennel Evangelista*. It misclassified *Hughes Electronics* and *MURDOCH SATELLITE* as persons, due to "Hughes" and "Murdoch" being listed as valid person first names followed by unknown proper names. We spuriously identified *MTV* and *CNN* as organizations, because they were both present in our company gazetteer and we made no attempt to disambiguate companies from TV channels. We also identified "March" in *the Long March rocket* as a date.

Some of these errors are easily corrected with a minor gazetteer additions. Experimentation to this effect lead to the correct classification of *Intelsat* and *Hughes Electronics*, and the avoidance of the spurious date "March". We also then correctly classified *Latin America* as a location, but the Namematch module then caused *Latin* and *Latin American* to also be classified as locations.

The remaining errors can be avoided by using the immediate context of the names, for example *Globo* is qualified by *conglomerate*, which, if added to our ontology of organization types, would allow *Globo* to be classified. Also, *Llennel Evangelista* could be classified correctly if the appositional phrase *a spokesman for Intelsat* were interpreted correctly.

Coreference

Task	Recall	Precision	P&R
CO Walkthrough	60.8	64.0	62.3
CO Overall	56.1	68.8	61.8

The coreference algorithm performed well on all definite NPs, and on the majority of proper names in the walkthrough text. Only half of the pronouns are correctly resolved, however, and this is less than the system was typically able to resolve on the training data. The main reasons are explained below. More generally, the walkthrough recall result is higher than the average recall we obtained with the training and the formal run data (respectively 56.5% and 56.1%). The precision result is lower than the average obtained on the training and the formal run data (respectively 72.3% and 68.8%). We concentrate below on the sources of errors that caused precision to drop.

The most common error is when a pronoun corefers to an entity that precedes it and has not been ruled out as a possible antecedent by possessing an incompatible syntactic or semantic property (the basic algorithm is 'eager', in the sense that it will corefer a pronoun with the closest preceding entity which cannot be ruled out as an antecedent). This happens for three pronouns in the walkthrough text. For example in the following paragraph, *its* corefers to *revenue* while the correct antecedent is *Galaxy VIII*, the main focus of the paragraph.

> *Hughes' Galaxy VIII(I) plan would use one satellite, which the company estimates will cost $230 million to build and launch. Hughes expects Galaxy VIII(I) will bring in $30 million in revenue in its first year and $58 million each year for the following 11 years, according to filings at the FCC.*

The same problem occurs with the pronoun *they* that corefers to *airwaves* instead of *functions*, in the following:

> *Those functions are likely to be slowly shifted to another slice of spectrum, while the airwaves they've historically used are turned over, in part, to satellite services such as the ones planned by GE and GM.*

This kind of spurious coreference is more likely to happen with words, like *revenue* or *airwaves* which are not recorded in our domain model, as no information is available about such classes to rule out or restrict coreference. Both examples suggest that a more complex mechanism is needed to detect which entity the paragraph is about, i.e. the *focus* or *center* (*Galaxy* in the first example, and *functions* in the second) and to constrain the pronouns to corefer with it. Substituting a focus-based approach based on [2], that provides such a mechanism for pronoun resolution, correctly resolves these three pronouns. However, such an approach has problems of its own: the result of applying the focus-based algorithm across the whole text shows a slight improvement for precision, but a somewhat larger drop in recall, due mainly to the fact that the focus-based approach is more restrictive in proposing antecedents for pronouns [3]. At present we are carrying out experiments to see how combining the advantages of both focus- and non-focus based approaches could lead to in superior results for pronoun resolution.

The mechanism which automatically adds nodes to the ontology for words of unknown semantic class, contributes strongly to recall. All the dynamically added nodes corefer correctly, i.e., the definite noun phrases *the allocation, the airwaves, the plan* whose head nouns were not known in our ontology are coreferred correctly.

Coreferring instances introduced by head nouns which are identical or of compatible semantic type is complicated when they have different qualifiers or modifiers. We have been handling these cases carefully in our coreference algorithm, but the walkthrough text provided some examples we had not met in the training data. Correcting our algorithm for the qualifier comparison (to avoid coreferring things like *direct-to-home service* and *direct-to-home video satellite service*) led to a small improvement: R: 62.0; P: 68.1; P & R: 64.9 on the walkthrough text, and R: 56.0; P: 70.2; P & R: 62.3 when run across the whole formal run test set.

Another source of error is when the parse extension mechanism (described in the section on Discourse Interpretation above) proposes the wrong argument for a verb. Such is the case in the example below, where *it* and *the allocation* do not corefer because the parse extension mechanism, when looking for a missing logical subject for *use*, hypothesises that it is *the allocation*; *the allocation* is then ruled out as an antecedent of the pronoun, since the logical subject and logical object of the same verb cannot corefer, except for particular cases such as the copula.

> *Other companies that support the allocation and may use it include Lockheed Martin Corp.'s Loral Space and Communications, International Private Satellite Partners/Orion Atlantic Capital Corp., and Comsat Corp.*

The system also failed to detect that the pronoun *it* that occurs in the pleonastic construction *It is ... critical* is not coreferential, and tried to corefer it anyway. Pleonastic constructions are addressed in our system; however, in this case the parser was thrown off by the ellipsis.

Finally, incomplete noun group recognition of *Hughes' Galaxy VIII(I)*, i.e., not including *(I)* as a part of the name, resulted in identifying *I* as a pronoun, then coreferring it with *Rupert Murdoch*.

Information Extraction

Texts/Task	TE			TR			ST		
	R	P	P&R	R	P	P&R	R	P	P&R
Walkthrough	75	87	80.68	39	77	51.97	42	40	41.18
Overall	75	80	77.17	41	82	54.70	47	42	44.04

Template Elements TE scores on the walkthrough article were slightly above the average across the test set. On this article we were largely successful with all slots except for the Entity Descriptor slot where scores were 50 % precision and 21 % recall. We will first explain the particular items we failed on, and then discuss why our Entity Descriptor slots were so poor.

The system performed relatively poorly on the three artifacts in the walkthrough text. We did get the *Long March 3* artifact, but missed the *B*, so the name was incorrect. We identified both of the artifacts without names – the Intelsat satellites – but incorrectly used *Intelsat* as the artifact name, and got the descriptors wrong.

On organizations the system was much better, failing on only four (of twenty-three). We did not corefer *News Corp.* and *News Corporation*, nor did we corefer *TCI* and *Tel-Communications Inc.* because the name matcher module did not equate them. Therefore an extra template element was generated for both *TCI* and *News Corp.* We also did not recognize *ING Barings* as a company, and so did not print a template for it. Finally, we reported *Organizacoes Globo* as a city instead of a company. Since it was not in our Gazetteer we did not know its type; it was adjacent to a country, so we guessed it was a city.

The system also did quite well on persons, failing on only three (of ten). We did not classify *Shayne McGuire* or *Virnell Bruce* as persons because their first names did not appear in our Gazetteer. We failed to get the unnamed *company spokesman* element, because we had made a conscious decision only to output templates for entities with a `name` field. This was done because our handling of descriptors was so weak.

The system also did quite well on locations, failing on only four (of nineteen). We failed to print *Brazil* since we printed *Organizacoes Globo* as a city with Brazil as its country. We failed to identify *Arlington* as a city because it was in the Gazetteer as an organization and thus was attached to *Space Transportation Association*; this led us to mistakenly print *Virginia* as a province. We printed *U.S.* from *U.S.-based* which seems correct. We simply missed *French Guyana* because it was not in the Gazetteer.

The system did quite poorly on Entity Descriptors getting only four right, two incomplete, two wrong, and missing thirteen. We had incomplete answers with *Televisa* where we got *Mexico* instead of *Mexico's biggest broadcaster*. This was due to our weak handling of the *'s* in the grammar. We were also incomplete with *Irving Goldstein* where the Descriptor proposed was *chief executive* instead of *director general and chief executive of Intelsat*. This was due to our poor handling of conjunction during parsing.

The system proposed two spurious Descriptors. First, we gave *Rupert Murdoch* the descriptor *group*, again due to a grammar confusion involving *'s*. Second, we gave *TCI* the descriptor *media* due to a problem with name lists.

The system also missed thirteen Descriptors altogether. This was largely due to a combination of a conservative approach to parsing and a lack of effort on filling the Descriptor field. Our basic approach to parsing was to only make attachments when we were quite sure they were correct. This meant that many of the complex Descriptor fields such as *spokesman for the Space Transportation Association of Arlington, Virginia* were never successfully combined during parsing. We left these attachments to the discourse interpreter, but we did not have enough time to develop discourse interpretation rules to make these attachments.

Improving scores on Entity Descriptors would have improved other aspects of our TE performance. We could not propose Elements based simply on Descriptors because our Descriptors were so poor. Furthermore, Descriptors can help to classify entities, as in the case of *Shayne McGuire, spokesman*.

Template Relations TR scores on the walkthrough text were slightly below the test set average. In the walkthrough text we correctly identified six of eight `location_of relations`, six of ten `employee_of` relations and zero of two `product_of` relations.

In no formal run text did the system correctly identify a `product_of` relation. We had only one discourse interpreter rule to find this relation and that relied on discovering a known aircraft manufacturer. While quite restrictive, this rule had led to high precision in the air crash domain of the dry run training data. We had believed that there would be a large degree of overlap between aircraft and rocket manufacturers but we were incorrect. Perhaps we would have been more effective on this problem if we had had some training data for Relations in the domain of launch events, but none was made available.

Of the other six relations that we missed, three are due to failing to correctly categorize template elements. That is, the TR task fails because the TE task fails. One `employee_of` relation is missed due to our conservative parsing strategy; we do not attach *for the Space Transportation Association* to *spokesman* in either the parser or discourse interpreter. One `location_of` relation is not generated due to our weakness of *'s* processing. The final `location_of` relation is not generated because we do not completely parse *International Technology Underwriters* and thus do not attach it to the adjacent location.

In addition to including a list of rocket manufacturers, two major improvements could be made to improve our TR scores. The first is to improve the TE scores, in particular by improving our Descriptors. Secondly we could improve our phrasal attachment either in the grammar or in the discourse interpreter. This would enable us to see more relations between entities that are nearby in the text.

Scenario Template The walkthrough text is reasonably representative of our overall performance on the ST task. The system proposed four launch events, twice as many as in the key, due to the identification of four separate satellites, each of which was associated with an event which then prevented event coreference.

One of the spurious satellites was mentioned as a qualifier: *an Intelsat satellite launch*. We attempted to avoid these cases with a rule to prevent any hypothesised entity resolving with noun modifier within a nominal compound. This assumes that all entities required for the template will be mentioned as head nouns at some point in the text. Unfortunately this rule failed altogether due to a bug.

The other spurious satellite was caused by a coreference failure. The coreference mechanism includes a strict rule that indefinite noun phrases do not have antecedents, and so no coreference was made between *an Intelsat satellite* and *a satellite built by Loral Corp. of New York for Intelsat*, resulting in a spurious launch event for the second case.

Our system's poor performance on descriptors also lead to incorrect payload entities for the satellite, despite identifying the correct instances from the input, e.g. *my satellite* as a descriptor for *a second Intelsat satellite*. This was corrected by introducing different criteria for substantial descriptors in the ST task than for the TE/TR tasks where we only output descriptors for entities that also had names.

Correcting the hypothesis resolution rule and the descriptor selection increased both recall and precision on this text, with a 4.5% increase in P&R.

Further errors included the system's failure to identify any FAILED launch events. All events proposed for the walkthrough text were SCHEDULED. We also associated two launch events with *the shuttle* as a vehicle, rather than *Long March 3B*, and we missed the *B* from the rocket name.

The system's (official, not debugged) ST output can be summarised as follows:

1. A civilian TV satellite is to be launched today from China.
2. A civilian TV satellite is to be launched in a year from the US by Long March 3.
3. A civilian TV satellite is to be launched in a year from China by the shuttle.
4. A civilian TV satellite is to be launched in a year from the US by the shuttle.

The production of these simple summaries in not yet automated, as in LaSIE-I for MUC-6, but we plan to add this facility in the near future. The construction of a complete discourse model by the system allows various results to be read off easily, not only the MUC output. For summarisation we therefore have access to much more information than that contained in the template itself, with the ability to control the level of detail. Such summaries can be used as a debugging tool, especially as a means for non-technical users to provide error reports.

DISCUSSION

Before closing, two questions are worth considering. First, how did our MUC-7 results compare with our MUC-6 results and what does this tell us? Second, what would we do differently or next in order to improve on our MUC-7 performance? These questions are best discussed in relation to each of the tasks.

Named Entity

Our official MUC-6 NE scores were R: 84;P: 94; P & R: 89.06. Our system failed to process two texts in the MUC-6 evaluation due to a bug totally unrelated to the system's language processing capabilities; scoring done by the MUC scoring committee with this bug fixed yielded R: 89;P: 93; P & R: 91.01 which is a more accurate reflection of the system's capabilities. This time our offical scores were R: 83;P: 89; P & R: 85.83 – a drop of about 5 %.

The first thing to note is that the scores of many of the MUC-6 veterans dropped comparably on NE in MUC-7. One obvious cause of this was lack of training data in the domain of the final evaluation. In our case, and in some other cases, one manifestation of this was the failure to recognise astronomical bodies as locations, since there were none, or very few, astronomical bodies in the NE dry run data. As noted above, adding this change, and one or two other small changes apparent from examining the NE test data (which subsequently became the IE training data), lead to improved scores of R: 87; P: 94; P & R: 90.41.

Thus, on balance, a case can be made for claiming our NE scores have remained broadly the same as in MUC-6. But, given the further effort that has gone into the system since MUC-6, one would have expected scores to have improved. Why haven't they? Without further detailed failure analysis we cannot say precisely. However, a few remarks can be made. First, it appears that the NYT texts used for MUC-7 are more heterogeneous in their style, and hence there is more variation in form of NE expressions. Designing a rule set to capture this variation is, consequently, more difficult. This observation has been supported anecdotally by variousMUC-7 participants. Second, the NE task was harder in certain respects. In particular, relative temporal expressions were included in the MUC-7 NE task – both for dates and times of day. Some of these expressions were very difficult indeed (e.g. *less than one hour after the American Airlines 757 slammed into the mountain side killing all 147 passengers*) and it is not clear that the task guidelines had completely stabilised for this subtask.

What would we do differently/next to improve NE performance? Most obviously, since our recall is considerably below precision, we need to concentrate on recall. Our system has a category of 'unknown' proper name (any proper name not resolved to one of the MUC categories) and even superficial reviewing of system results show that many proper name expressions falling into this category ought in fact to have been assigned one of the MUC NE categories. Given that the system carries out a relatively deep analysis, it should be possible to use further lexical semantic/conceptual knowledge in the discourse interpreter to resolve some of these cases. [2]

Another area that needs work concerns the determinism of the NE grammars. The ten NE grammars used in the MUC-7 system operate in a fixed order and the last co-spanning analysis always gets chosen. In some cases, regardless of the order of the grammars, errors will result. For example, consider the names *Julian Hill* and *Pearl Harbour*. As presented to our NE grammars both consist of a known person first name followed by a location trigger word. If our person grammar is run after our location grammar both come out as persons; if the location grammar is run after the person grammar both come out as locations. Clearly, both could be either locations or persons, though most of us will use world knowledge to make the choice. However, the best solution is to pass on the ambiguity and allow a later module, with more contextual knowledge, to decide. Controlled propagation of ambiguity into the discourse interpreter is thus a challenge for us.

Coreference

Our MUC-6 official coreference scores were: R: 51; P: 71 (P & R: 59.36[3]). As with NE we missed processing two texts and when the results for these were added in our scores were R: 54; P: 70 (P &R: 60.97). For MUC-7 our scores were R: 56.1; P: 68.8; P & R: 61.8.

Thus, there has been a very slight overall improvement. Considerable effort went into fine-tuning the coreference mechanism and more detailed failure analysis will be necessary to determine why more significant gains were not achieved and where further advances can be made. The final test set was smaller in MUC-7 (20 articles, as opposed to 30 in MUC-6) and it may be that the extra coreference phenomena we addressed in our MUC-7 system (see above) simply did not occur in the test set with sufficient frequency to make a significant difference. Or it may be that the MUC-7 CO test was significantly harder in some way that has not yet been determined.

Obvious areas for work include better handling of quoted speech, ascertaining whether the combination of a focus mechanism and a semantic compatibility/recency mechanism can yield overall better results, and improving the underlying grammatical analysis on which the coreference algorithm relies.

Information Extraction

Template Element For the template element task our official MUC-6 scores were: R: 66; P: 74; P & R: 69.8 and, with the addition of the two missed articles: R: 68; P: 74; P & R: 70.8. MUC-7 scores were: R: 75; P: 80; P & R: 77.17.

Template element is the one MUC-7 task where we appear to have made clear and significant gains over MUC-6. Of course, the major redefinition of this task since MUC-6 may mean direct comparison is not sensible. However, it is worth briefly considering whether a reasonable story can be told about why our scores have gone up.

[2]We did attempt to collect information from the training keys about the distribution of NE classes occurring as complements of specific prepositions (e.g. what sort of NE's follow the preposition *in*) and as noun phrases modified by PPs with a specific preposition and complement type (e.g. what sort of NE's are modified by phrases of the form *in* LOCATION). However, none of these patterns occurred reliably enough to be used in the final system.

[3]F-measures were not calculated for the coreference task in MUC-6, but were for MUC-7. The P & R figures supplied here for MUC-6 are our calculations using the standard formula.

Our NE scores were no better than in MUC-6 and TE is crucially dependent on the ability to identify locations and organizations. After analysis it appears that the increase is due to two factors. First, a change in the TE task definition meant that the location information associated with an organization in MUC-6 – the `locale` and `country` slots of the `organization` template element – was exported into the `location_of` template relation in MUC-7. Since we did relatively poorly in identifying this information in MUC-6 (relative to other slots in the `organization` template), it seems reasonable that our scores on TE would go up when it was removed from the task. Second, our `entity_descriptor` scores, while bad in MUC-7, were much worse in MUC-6: in fact these scores improved by a factor of three in MUC-7. We believe this is attributable to less ambitious, but more reliable phrasal parsing, and to the creation of a more finely tuned set of rules in the discourse interpreter specifically geared to extracting entity descriptors.

What would we do differently/next to improve TE performance? As noted in the discussion of the walkthrough text above, the `entity_descriptor` slot is where most effort needs expending, and to improve this we need more reliable PP-attachment in descriptive phrases (many of our descriptors were fragments of the correct descriptor) as well as more work on identifying which phrases are descriptors. Other weak points were low recall on artifacts and low precision on locations.

Template Relation The template relation task was new for MUC-7, so there are no MUC-6 scores to compare with (the information for one of the MUC-7 relations – `location_of` – was captured in the template element for `organization` in MUC-6, but it is not clear how to compare meaningfully the `locale` and `country` slot scores in the MUC-6 `organization` TE, with the MUC-6 TR `location_of` object and slot scores).

As discussed in relation to the walkthrough text above, the major improvements needed for TR are first to develop appropriate rules for `product_of` relations, since we missed all such relations in the final evaluation; second, to improve TE scores, since TR is parasitic on TE; and third, as with TE, to enhance grammatical and semantic analysis to better detect TR's.

Scenario Template Our MUC-6 ST scores were: R: 37; P: 73; P & R: 48.96 (the addition of the two missed articles made no difference as they contained no scenario events). In MUC-7 our ST scores were: R: 47; P: 42; P & R: 44.04. Thus, we suffered an overall drop in f-measure of nearly five points and while over recall went up by ten, our precision, which been the highest ST precision score at MUC-6, went down by over thirty points.

We believe that the MUC-7 ST task was significantly more difficult than the MUC-6 one. This conclusion appears to be borne out by the overall lower performance on ST in MUC-7 (high f-measure of 50.79 as compared to 56.4 in MUC-6) and is the result of a number of factors. First, the MUC-7 texts were longer on average and told more complex stories. Second, the template for MUC-7 was more complex, consisting of 7 object types with a total of 32 non-optional slots, while the MUC-6 template consisted of 5 object types with 20 non-optional slots. Third, the MUC-7 template required more fine-grained distinctions to be made. For example, no less than five organizations had to be distinguished: the launch vehicle owner, launch vehicle manufacturer, the payload owner, payload manufacturer and payload recipient. These are subtler distinctions than those required in the MUC-6 management succession task.

Our drop in precision was largely due to being unable to appropriately merge multiple references to the same event, as discussed in reference to the walkthrough article above. Our belief, though this requires further analysis, is that this is more of a problem in the MUC-7 ST task because the texts tend to refer to the same scenario event repeatedly more than the MUC-6 texts do. We also observed a tendency for events in the MUC-7 scenario to be expressed more frequently by nominalisation (*(rocket) launch*, *(missile) attack*, *(shuttle) flight*, even *(shuttle) mission* could all signal a `launch_event`) than the management succession events had been in MUC-6. Our attempts to handle these no doubt contributed to recall but had a large negative impact on precision, as accurately merging multiple nominalisations of the same event is difficult.

Without considerably more analysis of the results it is difficult to identify the most promising avenues to improve LaSIE-II's MUC-7 ST performance. As ever, more time to implement the scenario would have helped. We devoted under ten person days to this task and given its complexity the results are not surprising (we produced no fill at all for several slots, due to lack of time to implement any rules for them).

CONCLUSION

From our perspective the most encouraging result from MUC-7 was the vindication it supplied for the effort we have put into the GATE architecture over the two years leading up to the evaluation. The chief advantages GATE afforded were:

- a framework supporting a highly modular approach to language processing which in turn permitted a team with varying levels of programming skills, areas of expertise, and available time to work effectively together;
- reusability of interface code, especially results viewers, that allowed rapid creation of useful tools for gaining diagnostic insights.

Using GATE we were able to take part in all five of the MUC-7 tasks without excessive expenditure of effort – everyone working on MUC-7 had a parallel full time commitment to other projects. Further, components of the MUC-7 system were in active use in systems undergoing simultaneous development for other language engineering projects, and GATE made managing this complexity straightforward (this contrasts with LaSIE-I which was a monolithic system more or less dedicated to MUC-6).

As noted at the Introduction, LaSIE-II does not diverge radically from LaSIE-I in general approach, and as we did not set out to test explicitly any hypothesis about language processing in the evaluation, we do not see MUC-7 as allowing us to draw any strong conclusions confirming or disconfirming aspects of this approach. Perhaps the most interesting insights we have gained as a result of participating in MUC-7 are the following.

- Simple replacement of a semantic compatibility/recency approach to pronoun resolution with a focus-based approach does more harm than good – perhaps because the focus-based approach relies on more accurate/complete syntactic information than is available from a parser designed to perform robust syntactic analysis on real texts.
- Using a hand-crafted grammar which aims only to do phrasal analysis up to the point of ambiguity, as opposed to a grammar extracted from the PTB and thresholded on rule frequency (as we did for MUC-6), produces less complete but more accurate syntactic analyses. How this feeds through to task performance is hard to assess. More work needs to be done to see how and to what extent these partial syntactic analyses can be extended using conceptual knowledge.
- Further techniques need to be devised for (semi-) automatically acquiring and refining lexical semantic/conceptual knowledge in the domain.

MUC-7 was both a harder and a broader test than MUC-6, so we are neither surprised nor dismayed by the lack of striking progress in 'bottom line' figures. The data, task definitions, and scoring software produced for MUC-7 are a rich resource which we intend to mine for deeper insights for the foreseeable future. From these insights further progress is sure to follow.

ACKNOWLEDGEMENTS

This work was partly supported by EPSRC grant GR/K25267 (GATE/LaSIE), EC DGXIII grant LE 2238-1 (AVENTINUS), GlaxoWellcome plc and Elsevier Science (EMPathIE).

REFERENCES

[1] S. Abney. Partial parsing via finite state cascades. *Natural Language Engineering*, 2(4):337–344, 1996.

[2] S. Azzam. Resolving anaphors in embedded sentences. In *Proceedings of the 34th meetings of the Asssociation for Computational Linguistics (ACL'96)*, Santa Cruz, CA, 1996.

[3] S. Azzam, K. Humphreys, and R. Gaizauskas. Evaluating a focus-based approach to anaphora resolution. In *Proceedings of COLING-ACL'98*, pages 74–78, 1998.

[4] E. Brill. A simple rule-based part-of-speech tagger. In *Proceeding of the Third Conference on Applied Natural Language Processing*, pages 152–155, Trento, Italy, 1992.

[5] H. Cunningham, K. Humphreys, R. Gaizauskas, and Y. Wilks. Software Infrastructure for Natural Language Processing. In *Proceedings of the Fifth Conference on Applied Natural Language Processing (ANLP-97)*, pages 237–244, March 1997. Available as `http://xxx.lanl.gov/ps/9702005`.

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

[7] R. Gaizauskas and K. Humphreys. Using a semantic network for information extraction. *Journal of Natural Language Engineering*, 3(2/3):147–169, 1997.

[8] R. Gaizauskas, K. Humphreys, S. Azzam, and Y. Wilks. Concepticons *vs.* lexicons: An architecture for multilingual information extraction. In M.T. Pazienza, editor, *Proceedings of the Summer School on Information Extraction (SCIE-97)*, LNCS/LNAI, pages 28–43. Springer-Verlag, 1997.

[9] R. Gaizauskas, T. Wakao, K Humphreys, H. Cunningham, and Y. Wilks. Description of the LaSIE system as used for MUC-6. In *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, pages 207–220. Morgan Kaufmann, 1995.

[10] R. Grishman. The NYU system for MUC-6 or where's the syntax. In *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, pages 167–176. Morgan Kaufmann, 1995.

[11] R. Grishman. TIPSTER Architecture Design Document Version 2.3. Technical report, DARPA, 1997. Available at `http://www.tipster.org/`.

[12] P. Harrison, S. Abney, E. Black, D. Flickinger, C. Gdaniec, R. Grishman, D. Hindle, R. Ingria, M. Marcus, B. Santorini, and T. Strzalkowski. Evaluating syntax performance of parser/grammars of english. In *Proceedings of the Workshop On Evaluating Natural Language Processing Systems*. Association For Computational Linguistics, 1991.

[13] J.R. Hobbs, D. Appelt, M. Tyson, J. Bear, and D. Israel. Description of the FASTUS system as used for MUC-4. In *Proceedings of the Fourth Message Understanding Conference MUC-4*, pages 268–275. Morgan Kaufmann, 1992.

[14] 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.

[15] D. McKelvie, C. Brew, and H. Thompson. Using SGML as a Basis for Data-Intensive NLP. In *Proceedings of the fifth Conference on Applied Natural Language Processing (ANLP-97)*, 1997.

[16] R. Quirk and S. Greenbaum. *A University Grammar of English*. Longman, Harlow, Essex, 1973.

[17] T. Wakao, R. Gaizauskas, and Y. Wilks. Evaluation of an algorithm for the recognition and classification of proper names. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING96)*, pages 418–423, Copenhagen, 1996.