To appear in *Corpus-Based and Computational Approaches to Discourse Anaphora*, S. Botley and T. McEnery (eds). UCL Press and the Centre for Computational Linguistics, UMIST, 1996.

Quantitative Evaluation of Coreference Algorithms in an Information Extraction System

Robert Gaizauskas and Kevin Humphreys
Department of Computer Science
University of Sheffield
{robertg,kwh}@dcs.shef.ac.uk

Abstract

Algorithms for performing coreference resolution can only be precisely evaluated given a benchmark corpus of coreference-annotated texts, together with techniques for evaluating the algorithms' output against the corpus. Such a corpus and such techniques have become available for the first time as part of the Message Understanding Conference 6 (MUC-6) evaluations of information extraction systems. In this paper we describe the MUC-6 coreference task and the approach to taken to it by the Large Scale Information Extraction (LaSIE) system developed at the University of Sheffield. The basic coreference algorithm used by this system is described in detail, as well as a set of variants, which allow us to experiment with different constraints such as restrictions to certain classes of anaphor, distance restrictions between anaphor and antecedent, and weighting factors in assessing semantic similarity of potential coreferents. Quantitative evaluation results are presented for these variants, demonstrating both the utility of quantative analysis for assessing coreference algorithms and the flexibility of our approach to coreference which provides a framework that facilitates experimentation with alternative techniques.

1 Introduction

Information extraction (IE) is a term which has come to be applied to the activity of automatically extracting pre-specified sorts of information from short, natural language texts – typically newswire articles (see, e.g., [Jac92]). For instance, one might scan business newswire texts for announcements of joint ventures and extract the names and nationalities of the participating companies, the activity of the venture, the start date of the venture, its capitalisation, and so on. Put another way, IE may be seen as the activity of populating a structured information source (or database) from an unstructured, or free text, information source. In this process much is lost – all the information which does not fit into the 'template', the predefined structure which stipulates the sort of information to be extracted; but much is gained – a structured representation of large numbers of texts which can then be automatically searched and analysed in ways that human readers could not feasibly be expected to.

Research into the design and evaluation of IE systems has been stimulated by a series of competitive software system evaluations sponsored by the US Advanced Research Projects Agency (ARPA). These evaluations, known as *Message Understanding Conferences* (or MUCs), have been occurring roughly bi-annually since 1987 (see [CHL93], [GS96] for overviews and histories of some of the MUCs). The key feature of these events is that the extraction task is first precisely defined and then, while participants are developing automated systems to attempt the task, human analysts are employed to perform the extraction task manually, in order to create a test corpus of texts and corresponding filled templates, against which participants' automated systems may later be evaluated. The climax of the event is a final run during which the participants submit the texts in the unseen test corpus to a 'frozen' version of their system and then return the resulting system-generated templates to the organisers who score them against the manually extracted (and hence 'correct') templates. The result of the exercise is a set of quantitative evaluation figures which benchmark automated language processing techniques against human performance on the same, real world texts.

The MUC evaluations have become increasingly sophisticated. The first five MUCs concentrated on the core IE task of template filling, covering such diverse subject domains as naval command and control messages, newswire reports of terrorist attacks, of joint venture announcements, and of micro-electronic product annoucements. In MUC-5, the evaluation was broadened to include a language other than English (Japanese). However, in the most recent MUC (MUC-6, late 1995), a new level of refinement was introduced. In response to participants' desires to evaluate their systems at a level more fine-grained than end-to-end template-filling capability, a number of optional evaluations were introduced. These evaluations assess how well a system fares at subtasks which it was generally agreed are prerequisite for carrying out template filling to a high degree of accuracy. Candidate subtasks initially included word sense disambiguation, parsing, and predicate-argument identification, but constraints of time and the difficulty of agreeing on task definitions eventually reduced the non-template filling tasks to just two: the identification of named entity (NE) expressions (such as organisation, location, and person names) and, of most relevance here, a limited form of coreference identification.

Correctly identifying coreferences is of significance for IE because discovering template slot values frequently depends upon being able to follow coreference links. For example, to determine the corporate position of Dirk Ruthless from

> *Dirk Ruthless of MegaCorp made a stunning announcement today. In September he will be stepping down as Chief Executive Officer to spend more time with his pet pirhanas ...*

we must correctly resolve the pronomial anaphor "he" in the second sentence with "Dirk Ruthless" in the first. This phenomenon is pervasive in natural language text and it is hard to see how any IE system could achieve high levels of performance without the ability to identify coreferences.

The Natural Language Processing group at the University of Sheffield entered an IE system – the LaSIE (Large Scale Information Extraction) system – into all four tasks in the MUC-6 evaluation (two template filling tasks, named entity recognition, and coreference identification). In this paper we focus on the coreference task and on how LaSIE carries out this task. Given the nature of LaSIE's design, it should be seen not as embodying a fixed coreference algorithm, but rather as containing a base coreference algorithm on top of which various heuristics may be added or removed or combined to test their effectiveness. The presence of an evaluation benchmark, the MUC-6 annotated coreference corpus, allows us to carry out this testing precisely and quantitatively. We report on these tests here.

The paper is structured as follows. In section 2, we describe the MUC-6 coreference task in more detail, discussing the annotation scheme used to mark coreference relations, the class of coreference relationships which are to be marked up, the scoring algorithm, and the evaluation test corpus. In section 3, we give an overview of the LaSIE system as a whole, to provide a context for the coreference resolution. Section 4 describes the system's approach to coreference resolution in detail. First we discuss the formalisms used for representing the meaning of individual sentences and for representing the discourse and the background conceptual and world knowledge needed for robust interpretation. Then we introduce the coreference algorithms the system uses, firstly the base algorithm and then various heuristics which have been added to it. Section 5 presents the results of evaluating various configurations of the coreference algorithms. Section 6 presents our conclusions.

2 The Coreference Task in MUC-6

The full and precise definition of the MUC-6 coreference task is presented in Coreference Task Definition v2.3 [MUC95]. The following is a synopsis of the core parts of that definition and borrows heavily from it, including many examples. It should, we hope, be sufficient for understanding the rest of this paper. The reader should keep in mind that this definition in no way purports to exhaustively describe the coreference phenomena in natural language, that it is concerned primarily with a certain sort of text – *Wall Street Journal* articles, and that some decisions were taken more or less arbitrarily in order to make the definition precise enough for computer scoring and to arrive at a definition within the time frame allocated for the MUC-6 evalation. Tremendous debate took place amongst the participants about this definition; the debate will no doubt continue, and the definition be further refined.

2.1 The Annotation Scheme

Coreferential expressions are annotated by adding SGML [Gol90] tags into the text. Given an antecedent `A` and an anaphor `B`, where both `A` and `B` are strings in the text, the basic coreference annotatation has the form

```
<COREF ID="100"> A </COREF> ... <COREF ID="101" TYPE=IDENT REF="100"> B </COREF>
```

So for example *Galactic Enterprises said it would build a new space station before the year 2016* would be marked up as

```
  <COREF ID="100"> Galactic Enterprises</COREF> said <COREF ID="101" TYPE=IDENT
  REF="100"> it </COREF> would build a new space station before the year 2016.
x
```

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

Two other attributes were included within `COREF` tags in the manually marked up corpus. The `MIN` attribute was used to identify the minimum string that would be accepted for 'full points' by the scoring algorithm – either the head of the phrase or a named entity. For example if *Galactic Enterprises Inc. of Gotham City, Lancs.* is later referred to as *Galactic Enterprises* then the following annotation would be adopted:

```
<COREF ID="100" MIN="Galactic Enterprises Inc."> Galactic Enterprises
Inc. of Gotham City, Lancs.</COREF>
... <COREF ID="101" TYPE="IDENT" REF="100"> Galactic Enterprises Inc.</COREF>
```

Full credit was given if any string including at least the `MIN` string and at most the full string was identified. This was to attempt to decouple the coreference task from the task of accurately parsing noun phrases.

The final attribute was a `STATUS` attribute which could only take the value `OPT` (optional) and allowed the analysts to indicate coreferences about which there was genuine uncertainty (e.g. use of a nickname with which the text's author presumed familiarity which the analysts did not have and could only guess). Systems were only scored on optional coreferences if they attempted them.

2.2 Definition of the Task

Coreference relations were marked between strings of certain syntactic categories only – nouns, noun phrases, and pronouns. Only some strings of these categories were annotated and these string classes were termed *markables*. Strings which were markable were annotated only if the thing to which they referred or which referred to them was also markable (so, e.g., a pronoun referring to a clause would not be markable). Examples of markables are:

- names and named entities (as defined in the MUC-6 named entity task) – e.g. the *Galactic Enterprises* example above;
- present participles modified by nouns or adjectives – e.g. *deficit financing*;
- pronouns (personal, demonstrative, possessive and reflexive forms) – e.g. in

  **He* shot *himself* with *his* revolver.*

  all of "He", "himself" and "his" should be marked coreferential.
- 'bare' nouns occurring as prenomial modifiers – e.g. in

  *Sheffield's production of *steel* has dropped due to foreign competition in the *steel* industry.*

  the two occurrences of "steel" should be marked.

Examples of non-markables are:

- names embedded in other names – e.g. the two instances of "Kent" in

  The Duchess of Kent might summer in Kent.

  are not marked;

- gerunds – e.g. in

  > _*Leaping over tall buildings* may be fun, but *it*'s also dangerous._

  the two starred expressions should not be marked.

- implicit pronouns – e.g. in

  > _John posted the letter and walked home._

  the implicit subject of "walked" should not be linked to "John" by marking an empty string (so gaps are never marked);

- conjoined noun phrases – e.g.

  > _*The boys and girls* enjoyed *their* breakfast._

  is not marked – unless there is separate coreference as in

  > _*John Doe* and *Jane Deer* decided to eat. *John* ordered steak and *Jane* ham._

  where "John" in the second sentence and "John Doe" in the first would corefer, and likewise "Jane" and "Jane Deer".

Given the definition of markable, the task definition identifies a set of coreference relationships to annotate. These are:

1. **basic coreference** Two markables that refer to the same object, set or activity are to be linked.

2. **bound anaphors** Links are made between noun phrases and anaphors bound by them even if they are not coreferential in the usual sense. E.g.

   > _*Every student* discovered *their* grades._

3. **apposition** Appositional phrases in which both noun phrases are definite and which are explictly marked via overt punctuation are marked. E.g.

   > _*John Major*, *the Prime Minister*,. . . ,_

   but not

   > _*Bloggs*, *an old friend of mine*_
   > _Treasury spokesman* *Jones*_

4. **predicate nominals and time-dependent identity** Predicative nominals are marked provided they are definite (regardless of time). So

   > _*Major* is *Prime Minister of Great Britain*._
   > _Thatcher* was *Prime Minister of Great Britain*._

   are both marked. But

   > _*Blair* might be *Prime Minister of Great Britain*._
   > _Politics* is *a profession for rogues*._

   are not marked.

5. **types and tokens** Coreference links are to be marked between two markables if they both refer to sets and the sets are identical, or if they both refer to types and the types are identical. The distinction between sets and types is not always easy to define and in cases where there is residual doubt the links are marked as optional. For instance, in

> *. . . *producers* don't like to see a hit wine increase in price . . . *Producers* have seen this market opening up and *they*'re now creating wines that appeal to these people.*

the three starred markables, if taken as referring to the same sets, would not be marked as coreferential since the set of producers who have seen the market opening up is presumably not the same as the set of those who have created new wines in response to this. However, these markables are taken has referring to the same *type* and hence are marked as coreferential.

6. **functions and values** An expression may refer to the value of a function at certain arguments by mentioning the function and arguments explicitly, by assuming the arguments implicitly from context, or by simply stating the value. In

   > *GM announced *its third quarter profit*. *It* was *$0.02*.*

   all three starred expressions are marked as coreferential. In

   > **The temperature* is *90* . . . The temperature is rising.*

   the first occurrence of "The temperature" refers to the value of the function at arguments whose value is supplied by context and that value is 90. Hence the first two starred expressions are marked as coreferential. The second occurrence of "The temperature" refers to the function (indirectly by reference to its first derivative) and not to its value and hence is not marked as coreferential with either of the earlier two expressions.

7. **metonymy** Metonymy is viewed as type coercion. For example, in

   > **The White House* held a press conference today. *The beleaguered administration* was defending its record on . . .*

   the White House is coerced to the administration operating out of the White House. Metonymical markables such as this are marked as coreferential if the entities referred to *after* coercion are identical. Thus, in the preceding example the two starred references are marked as coreferential. However, in

   > *I bought the New York Times this morning. I read that the editor of the New York Times is resigning.*

   the first reference to the New York Times is coerced into a copy of the paper published by the New York Times, while the second is coerced into the organisation; in this case no coreference is marked.

2.3 Scoring

Systems' results, called *responses*, are scored against manually marked up texts, called *answer keys*, or just *keys*. The measures used are variants of the standard *recall* and *precision* measures used in evaluating information retrieval systems (see, e.g., [Sal89]). Recall is a measure of how many of what a system was to find it actually found, precision a measure of how many of what the system found it was meant to find. For example, suppose for a given task there are 100 items to retrieve and a system retrieves 75, of which 50 are correct. Then its recall is 50/100 or 50% and its precision is 50/75 or 66.6%.

In the coreference task, a problem arises which requires that these measures be specially adapted. Clearly, more than two markables may corefer, i.e., there may be chains of coreferences, not simply coreferential pairs. In the case of chains, how to record the chain and how to score systems which fail to discover all the links in the chain become central issues. For example, suppose A, B, and C are coreferential. This fact could be recorded by links from both B and C to A, or by a link from B to A and one from C to B, or in several other ways (for the purposes of the task, the coreference relation is supposed to be symmetric and transitive). If a system response records these links one way and the answer key records them in another then this should not result in a penalty to the system. Further, if the system fails to record a link in a chain then some care must be exercised in assigning it a score. Suppose an answer key contains the links A-B, B-C, and C-D. If a system response discovers the links A-B and C-D, what score should it receive ? Intuitively it seems that precision should be 1 (two links are found and both are correct) and recall should be 2/3 – any specification of the linking between four identical entities will require three links and the system has found two links that are correct. Techniques for generalising these intuitions based on equivalence classes of coreferred entities have been worked out with the consequence that systems may specify links in an order-independent way, and be sensibly scored for partial results. See [VBA+95] for a full discussion of the definitions of precision and recall for the coreference task.

2.4 The Test Collection

The test collection consisted of thirty articles from the *Wall Street Journal*. These ranged in length from 83 to 1349 words and averaged 462 words. The total number of coreference links in the test corpus, as used in the tests reported here, was 1627.

3 *LaSIE* system overview

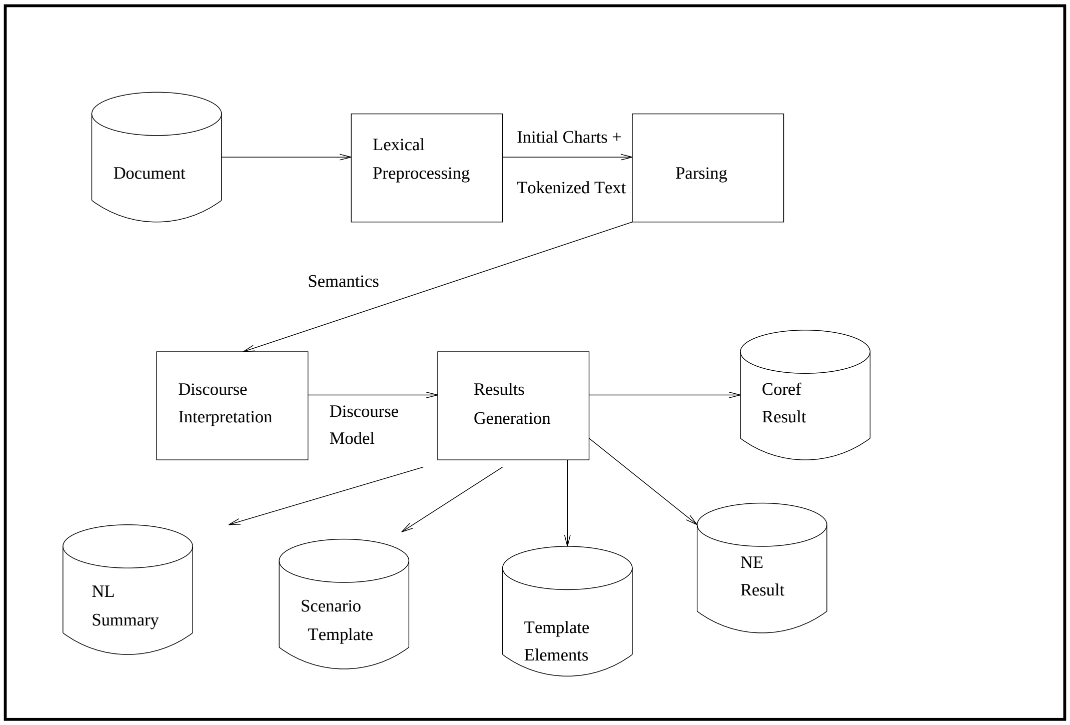

Figure 1: LaSIE System Architecture

LaSIE has been designed as a general purpose IE research system, initially geared towards, but not solely restricted to, carrying out the tasks specified in MUC-6: named entity recognition, coreference resolution, template element filling, and scenario template filling tasks (see [MUC95] for further details of the task descriptions). In addition, the system can generate a brief natural language summary of the scenario it has detected in the text. All of these tasks are carried out by building a single rich model of the text – the discourse model – from which the various results are read off.

The high level structure of LaSIE is illustrated in Figure 1. The system is a pipelined architecture which processes a text sentence-at-a-time and consists of three principal processing stages: lexical preprocessing, parsing plus semantic interpretation, and discourse interpretation. The overall contributions of these stages may be briefly described as follows:

- lexical preprocessing reads and tokenises the raw input text, tags the tokens with parts-of-speech, performs morphological analysis, performs phrasal matching against lists of proper names, and builds lexical and phrasal chart edges in a feature-based formalism for hand-over to the parser;

- parsing does two pass chart parsing, pass one with a special named entity grammar, pass two with a general grammar, and, after selecting a 'best parse', passes on a predicate-argument representation of the current sentence;
- discourse interpretation adds the information in its input predicate-argument representation to a hierarchically structured semantic net which encodes the system's world model, adds additional information presupposed by the input to the world model, performs coreference resolution between new instances added and others already in the world model, and adds information consequent upon the addition of the input to the world model.

For further details of the system see [GWH$^+$95].

4 Co-Reference in LaSIE

4.1 The World Model

The discourse interpretation stage of LaSIE is based around the XI knowledge representation language [Gai95]. The language allows a straightforward definition of cross-classification hierarchies, the association of arbitrary attributes with classes or individuals, and the inheritance of these attributes by individuals.

In LaSIE, XI is used to represent a simple *ontology* of classes or 'concepts' directly relevant to the various MUC tasks. The ontology currently contains only 80 concept nodes but, as described below, new nodes may be created dynamically during processing. The manual development of the ontology for the MUC domain was not therefore a major task. Much of the initial ontology was derived directly from the MUC task specifications, ensuring that distinctions required in the IE template slots and the NE classifications were reflected in the ontology.

In MUC-6, the template filling tasks were to do with extracting information concerning *management succession events* from financial newswire articles. So, details about persons, posts, and organizations, and also about events involving persons leaving or taking up posts in organisations needed to be extracted. The higher levels of the ontology for this task have the following structure:

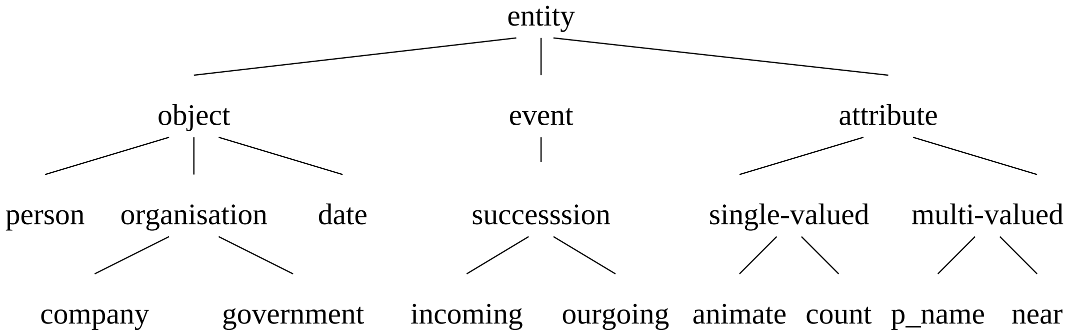

Figure 2: LaSIE Ontology

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

4.2 The Discourse Model

In addition to concept nodes, an ontology may contain instance nodes and these too may have associated attribute-value structures. However, in the MUC-6 application, the background, persistent LaSIE world model, that is the world model from which the processing of each text begins anew, does not contain any instance nodes. Instances are added only as the text is processed. Thus, the world model described above can be regarded as an empty shell to which the semantic representation of a text is added, populating it with instances mentioned in the text. The world model which results is a model specialised for the world as described by the current text; we refer to this specialised model as the *discourse model*.

Information about these instances comes from a predicate-argument representation, or quasi-logical form, produced by the parser as it processes the text, sentence by sentence. This representation describes the instances mentioned in each sentence, by identifying their semantic classes (derived directly from the syntactic form in most cases), and the relations which hold between them. For instance, the sentence:

ABC Inc. named Smith as its new chairman.

would have a predicate-argument representation of the following form:

```
company(e1), proper_name(e1, ABC Inc.),
name(e2), tense(e2, past),
proper_name(e3, Smith),
subj(e2, e1), obj(e2, e3),
pronoun(e4, its),
chairman(e5), adj(e5, new),
of(e5, e4)
```

Here the instance `e3` has a `proper_name` attribute but no semantic type, due to the lack of any syntactic information which would have allowed the named entity level parsing to classify the name 'Smith'. Similarly, the semantic type of `e4` is undetermined at this point. The sentence level parse also fails to cover the preposition 'as' (in this example) and so this relation is missed altogether from the predicate-argument representation.

All instances also have `realisation` attributes which specify the text from which the instance was derived, in terms of a range of text tokens, the sentence and paragraph in which it occurs, and whether it is part of the header or main body of the text.

The predicate-argument representation is processed by adding instances, together with their attributes, to the discourse model. On each addition, the model is checked for any inheritable `presupposition` attributes, the values of which are used to add (or remove) further information in the model. For instance, a `presupposition` attribute is associated with the node in the ontology corresponding to the `proper_name` attribute. When attempting to add `proper_name(e3,Smith)` to the model the `presupposition` attribute specifies that `e3` must be an instance of the class `object`, in the absence of any more specific information (i.e., the default semantic type of named entities is `object`, as opposed to, say, `event`).

Instances which have their semantic class specified in the input are added directly to the discourse model, if the class exists as a node in the ontological hierarchy. Again, the values of any inheritable `presupposition` attributes are established and applied to the model. If, however, the class specified in the input does not exist in the hierarchy, a new node is created dynamically. `event` instances in the input, such as `e2` in the example above, are distinguished from `object` instances by the presence of event-like attributes, i.e. `tense`, `subj` or `obj`, thus allowing a high level categorisation of unknown classes, which, in turn, allows potential coreferences to be established among instances of a class not originally present in the ontology.

4.3 The Base Co-Reference Algorithm

Following the addition of the instances mentioned in the current sentence, together with any presuppositions that they inherit, the coreference algorithm is applied to attempt to resolve, or in fact merge, each of the newly added instances with instances currently in the discourse model. Coreference resolution is performed by comparing the following sets of instances in this order. Coreference is only attempted between object instances, i.e. instances introduced by nouns, since references to events are outside the definition of the MUC coreference task.

1. compare: each instance mentioned in the current sentence using a proper noun
   with: every other instance in the discourse model which was mentioned using a proper noun

2. compare: each instance mentioned in the current sentence
   with: every instance before it in the current sentence

3. compare: each instance mentioned in the current sentence using a pronoun
   with: every instance mentioned in the current paragraph[1]

4. compare: each instance mentioned in the current sentence using a 'normal' noun (i.e. not a proper noun or pronoun)
   with: every instance mentioned in the current or previous paragraphs

These comparison sets effectively embody distance restrictions on the potential coreferences of the various noun types: proper nouns have no distance restriction, pronouns can only refer within the same paragraph, and normal nouns can only refer within two paragraphs. This last restriction on normal noun coreference was introduced mainly for reasons of efficiency, limiting the size of the comparison set when processing large texts.

Each comparison set may be viewed as a set of *candidate sets*, a candidate set being a set of pairs of instances all of whose first elements are the same (an instance in the current input) and whose second elements are possible instances, or candidates, occurring earlier in the text with which the first element might corefer. The algorithm proceeds as follows. For each pair of instances in each candidate set in each of the comparison sets listed above [2]:

1. Ensure semantic type consistency
   The semantic types of the two instances must be ordered in the ontology. If this is true a semantic similarity score is calculated using the inverse of the length of the path (measured in nodes) between the two classes. The attempt to resolve the two instances is abandoned if the semantic types are not ordered. For example, in the fragment of the MUC-6 ontology in Figure 2, `person` and `company` are not ordered with respect to each other in the ontology and therefore no pair of `company` and `person` instances would ever be coreferred. An instance of type `company` could be coreferred with one of type `organisation` or with one of type `object`; other things being equal, the former pair would be preferred on the grounds of higher semantic similarity.

2. Ensure non-distinctness
   Any additional coreference constraints are checked at this point to ensure that the pair of instances currently being considered do not possess any characteristics which imply that they should not be resolved. For instance, one of the constraints specifies that a new instance which has been introduced by an indefinite noun phrase in the text, should not be permitted to refer to any existing instance. The constraints are represented via the `distinct` attribute of certain nodes in the ontology, and should the current pair of instances inherit this attribute, the attempt to resolve them is abandoned. The various constraints currently implemented are discussed in the following section.

3. Ensure attribute consistency
   The values of any fixed single-valued attributes (as classified in the ontology, e.g. `animate`) common to both instances, must be identical. The attempted resolution is abandoned if any conflict is found.

4. Calculate a similarity score
   The semantic similarity score is summed with an attribute similarity score to give an overall score for the current pair of instances. The attribute similarity score is established by finding the ratio of the number of shared multi-valued attributes with compatible values, against the total number of the instances' attributes. If the `proper_name` attribute is among those shared, a name matching routine, specific to particular semantic types (i.e. person, organisation or other) is used to establish compatibility, and, if successfully matched, the attribute similarity score is strongly weighted to increase the overall score.

[1] The previous paragraph in the case of an initial pronoun if the current sentence starts a new paragraph.

[2] While the order in which instance pairs within a candidate set are examined cannot affect outcome of the algorithm, the order in which the candidate sets of a given comparison set are processed may indeed do so. We have not yet done any testing to determine just how significant this effect may be.

After each pair in a candidate set of a comparison set has either been assigned a similarity score or has been rejected on grounds of inconsistency, the highest scoring pair (if any score at all) are merged in the discourse model. If several pairs have equal similarity scores then the pair with the closest realisations in the text is preferred.

The merging of instances involves the removal of the least specific instance (i.e. the highest in the ontology) and the addition of all its attributes to the other instance. This will result in a single instance with more than one `realisation` attribute, which corresponds to a single entity mentioned more than once in the text, i.e. a coreference as required by the MUC task.

After all sentences have been processed, all instances in the discourse model with multiple `realisation` attributes are found, and the values of these attributes used to mark up the original text with SGML to indicate the coreference chains found by the algorithm. This marked up text can then be evaluated via the MUC scoring procedure.

4.4 Additional Constraints

The constraints on coreference represented via the `distinct` attribute act to rule out the potential coreference of an instance pair which may otherwise be permitted by the base algorithm. The constraints used in LaSIE for the final MUC evaluation were established through training on the coreference data provided for the MUC dry-run evaluation. Unfortunately not all the revisions made to the coreference task definition between the dry-run and final evaluations were allowed for in the LaSIE system, and so, with hindsight, not all the constraints used were entirely appropriate, as revealed by the evaluations in section 5.

The basic set of constraints as used in MUC-6 are as follows:

1. Prevent indefinite nouns from referring backwards
   A new instance introduced using an indefinite determiner is defined as being distinct from all other instances in the various comparison sets considered by the base algorithm, i.e. all instances before it in the text. For example, the phrase "an American company" would not be permitted to refer to any company previously mentioned in the text, reflecting an assumption that all indefinite determiners are used to introduce instances into a text for the first time.

2. Prevent non-pronouns from referring back to pronouns
   New instances mentioned using either proper nouns or full nouns are distinct from earlier instances which have been mentioned only by pronouns, i.e. preceding pronouns which could not be resolved with anything. An unresolved pronoun is thus prevented from being used as the root of a coreference chain in a text — roots must always be proper nouns or full nouns.[3]

3. Prevent unclassified proper names from referring back to dates
   New instances with `proper_name` attributes but with a semantic class no more specific than `object` are defined as distinct from all instances with a semantic class of `date`. This reflects the assumption that the recognition of date proper names at the earlier stages of processing is complete and correct, and so an unclassified name must be of some other semantic type.[4]

4. Prevent non-proper nouns used as qualifiers from coreferring
   An instance introduced as a qualifier or modifier of another instance is distinct from all other instances. Thus, no instance is permitted to corefer with the instance of the class `video` mentioned in the phrase "the video manufacturers". Unfortunately this constraint rules out a class of coreferences which are explicitly included in the final MUC coreference task definition. However, on the dry-run evaluation data, the constraint produced a useful increase in precision and so was retained for this reason.

5. Prevent pronouns from referring back to dates, numbers or locations
   This constraint is probably the most domain specific of those used in LaSIE. An apparent feature

[3] A single exception to this constraint is allowed: a noun which is the object of the verb *to say* can refer back to a first person pronoun, as in " '*I* agree', said *the chairman*". Clearly there will be generalisations of this case, but these should more properly be covered via a specific treatment of quoted speech, which is lacking in the current system.

[4] In fact LaSIE's performance on date expressions in the MUC named entity task was 94% recall, 97% precision, for the 30 texts common to the named entity and coreference tasks.

of financial texts is that repeated references to particular instances of dates, numbers or locations are rarely made, especially pronominal references. They are therefore disallowed altogether in LaSIE. However, the adverbs *there* and *then*, which may more commonly refer to dates and locations, are not treated specially at present.

The above constraints, applied via the `distinct` attribute, are all associated with the `object` node in the ontology, and are therefore inherited by all instances considered by the base coreference algorithm. The following two constraints are associated with the `date` node only:

6. Prevent proper noun dates from referring back to non-proper noun dates
   A new instance classified as a date by the previous named entity recognition stages, is distinct from all preceding dates without proper names. The assumption here is that a date will not be introduced as a normal noun phrase, such as "the month", and then later referred to by its full form, e.g. "January".

7. Prevent non-proper noun dates without definite determiners from referring backwards
   All new instances of dates which were introduced using normal nouns with no definite determiner, are distinct from all preceding date instances. For example, the instances of the class `year` derived from the phrases "years ago" or "23 years old" would not be permitted to corefer with anything, whereas the constraint would not apply to the instance derived from the phrase "this financial year".

The use of the `distinct` attribute provides a mechanism by which a wide variety of coreference restrictions can be expressed. Those listed above were all that were used in LaSIE for the MUC-6 evaluation, but some development has continued since. A further constraint has been added based on the identification of 'pleonastic' or 'non-referential' instances of the pronoun *it*, as proposed in [LL94]. Although the identification makes reference to purely syntactic and lexical information it can still be expressed via a `distinct` attribute of the `object` node in the ontology.

Lappin and Leass' test for pleonastic pronouns involves the recognition of patterns such as "It is **Modaladj** that **S**", where **S** is a sentence complement and **Modaladj** is a member of a set of lexical items such as *possible*, *useful*, *important*, etc. Such syntactic patterns can be identified within the discourse model in LaSIE, due to the preservation of much predominantly syntactic information via instance attributes in the semantic representation. For example, the above syntactic structure would have a predicate-argument representation of the following form:

```
pronoun(e1, it),
be(e2), tense(e2, present), subj(e2, e1), obj(e2, e3),
adj(e3, important)
```

where `e3` is the `event` instance described by the (verbal) head of the complement **S**. This allows, to a certain degree, the reconstruction of the original syntactic form from the semantics, providing a mechanism by which syntactic constraints can be expressed in the world model. The identification of syntactic patterns is, however, very much dependent on the performance of the parser and the grammar, and, as yet, their limitations have not been fully established.

5 Evaluation

LaSIE's performance in the MUC-6 trials was scored in the final evaluation as follows:[5]

Recall	Precision
801/1478 = 54.19%	801/1147 = 69.83%

These figures show the number of coreferences correctly identified by the system against, for Recall, the number of target coreferences in the manually annotated corpus, and against, for Precision, the total number of coreferences proposed by the system.

[5] In fact this is an unofficial score which includes the results from one text which LaSIE failed to process at all (for uninteresting reasons) in the official run. The official score had 3.68% lower recall, and 0.96% higher precision, because of the missed text.

Seven systems took part in the coreference task, with recall scores ranging from 35.69% to 62.78%, and precision scores ranging from 44.23% to 71.88%. LaSIE's score had the median recall of the seven systems and the second best precision. It should be noted that human inter-annotator consistency only significantly surpassed 80% after many refinements to the task definition.

The official evaluation involved the use of the scoring software in an 'interactive' mode, allowing judgements of borderline and optional cases to be made manually. Using the scorer non-interactively, as it is configured at Sheffield, the results produced from the same data were:

Recall	Precision
825/1627 = 50.71%	825/1147 = 71.93%

All the results presented below were produced using the scorer non-interactively in the same configuration as above. The relation of these results to official MUC scores would therefore be expected to show a similar increase in recall and drop in precision, although this has not been definitely established.

5.1 Variations on the Base Algorithm

For the following comparisons we take as our base system a slightly enhanced version of LaSIE as used in MUC-6. This version uses the base algorithm as described in section 4.3, and includes all the additional constraints described in section 4.4. The performance of this system is:

Recall	Precision
850/1627 = 52.24%	850/1188 = 71.54%

Restricting the system to proper name resolution (i.e. name matching) only, the results are:

445/1627 = 27.35%	445/501 = 88.82%

This shows that just over half of the coreferences found by the full system involve proper names only, and therefore could probably be identified by our name matching algorithms alone.

Restricting the system's ontology to the 80 or so predefined nodes, i.e. preventing the dynamic creation of any new nodes, the performance is:

556/1627 = 34.17%	556/1210 = 70.33%

Around 300 of the 850 coreferences correctly identified by the full system therefore involve semantic classes previously undefined in the ontology but which can simply be assumed to be subclasses of the `object` class.

A further set of tests was also carried out to investigate the effects of varying the instance similarity score calculation at step 4 of the base algorithm. However, no noticeable differences were found apart from the use of simply the number of shared consistent multi-valued attributes as an attribute similarity score, rather than the ratio of this number to the total number of properties. This variation produced performance increases of approximately 1.5% recall and 2% precision, one of the few variations giving an increase in both scores simultaneously.

5.1.1 Variations in distance restrictions

The base system for the following comparisons is again the enhanced MUC-6 system (R: 52.24%, P: 71.54%). The distance restrictions it incorporates are as described in section 4.3, i.e. pronoun antecedents must be within the same paragraph, and normal noun antecedents must be within the last two paragraphs. No distance restriction was imposed on proper nouns in any of the following tests.

Antecedent in:	Recall	Precision
current paragraph only	48.92% (-3.32%)	73.03% (+1.49%)
current + previous paragraphs	53.10% (+0.86%)	70.65% (-0.89%)
current + all previous paragraphs	56.73% (+4.49%)	65.83% (-5.71%)

Hobbs' results from the manual analysis of a corpus [Hob78] suggest that 98% of antecedents for the pronouns *he*, *she*, *it* and *they* are within the last two sentences. However he points out that "there is no useful limit on how far back one need look for the antecedent". A series of tests was run with LaSIE's distance restrictions varied in sentence units to investigate the applicability of Hobbs' result to a wider range of anaphors.

Antecedent in:	Recall	Precision
current sentence only	41.92% (-10.32%)	75.36% (+3.82%)
current + previous sentence	50.34% (-1.90%)	73.13% (+1.59%)
current + 2 previous sentences	51.81% (-0.43%)	71.62% (+0.08%)
current + 3 previous sentences	53.04% (+0.80%)	70.39% (-1.15%)
current + 4 previous sentences	53.47% (+1.23%)	69.54% (-2.00%)
current + 5 previous sentences	54.27% (+2.03%)	69.15% (-2.39%)

This shows that at least the previous three sentences must be considered to avoid losing any recall, when compared with the full system. Hobbs, however, does not explicitly consider paragraph units, as is possible here, and this may be more appropriate for certain classes of anaphoric references. This issue requires further investigation.

5.2 Additional Constraints

A test of the base algorithm alone, with all constraints removed, provided the following comparison:

System Configuration	correct	Recall	proposed	Precision
all constraints	850	52.24%	1188	71.54%)
no constraints	888	54.58% (+2.34%)	1355	65.53% (-6.01%)

The combined effect of the constraints described in section 4.4 is therefore to prevent the resolution of 167 instance pairs, 38 of which (i.e. 22.75%) were in fact correct.

Further tests were then run to establish the effects of the constraints individually. None of the constraints related to instances of the class `date` (constraints 3, 6 and 7 in section 4.4) had any noticeable impact on the MUC-6 test set. The constraint to avoid resolutions involving pleonastic *it*s, based on Lappin's and Leass' test as described in section 4.4, also had very little effect on this corpus. The constraint avoided the proposal of six spurious instance resolutions, producing an overall increase in precision of 0.35% without affecting recall.

The performances of the more effective constraints are shown below. These results were obtained by selectively disabling each constraint from a base system which included constraints 1–7, as described in section 4.4, but did not include the test for pleonastic *it*s.

Disabled Constraint	correct	Recall	proposed	Precision
base system	850	52.24%	1194	71.19%)
1. indefinite nouns	851	52.30% (+0.06%)	1210	70.33% (-0.86%)
2. non-pronouns→pronouns	855	52.55% (+0.31%)	1249	68.45% (-2.74%)
4. qualifier nouns	884	54.33% (+2.09%)	1257	70.33% (-0.86%)
5. pronouns→numbers	844	51.87% (-0.37%)	1195	70.62% (-0.57%)

From these results the removal of constraint 4., preventing coreferences of qualifier nouns, would give a reasonable improvement in recall without a great loss of precision. This reflects the fact that the constraint actually conflicts with the final MUC task definition, as discussed in section 4.4, and so such coreferences will be present in the manually annotated texts. Constraint 2., preventing non-pronouns from referring back to pronouns, is the most effective at avoiding spurious coreferences: 55 instance resolutions were prevented, only 5 of which should have been retained.

No analysis of the MUC-6 test corpus has been performed to identify the relative frequencies of the various classes of anaphors, and other characteristics such as the maximum distance between an anaphor and its antecedent, but clearly this information would allow a more focussed set of system variations, enabling a more detailed identification of current problems and possible solutions.

6 Analysis and Concluding Remarks

The results presented above demonstrate that while LaSIE's performance in the MUC-6 coreference task was above average, it was not optimally configured for the final evaluation, and indeed not all coreference classes in the task definition had been fully dealt with. Several of the constraints on the base algorithm have either no or a detrimental effect on overall performance, and should therefore have been omitted.

While LaSIE's precision score was quite good (especially given human performacnce on the task), its recall score stands in obvious need of improvement. Constraints, as we have considered above, only ever prevent potential resolutions suggested by the base algorithm, and yet the base algorithm still only achieves a level of recall around 55%. The reasons for this were sought by looking in detail at the MUC-6 system output for the first time. Prior to the test runs described in this paper the MUC-6 data had been kept 'blind' to avoid tuning the system to any particular characteristics of the data set.

The immediately noticeable problems were often related to errors and omissions in the predicate-argument representation passed on from the parser to the discourse interpretation stage. These stem, ultimately, from limitations in the grammar. Many cases of apposition, relative clauses and coordination were not parsed correctly or completely, producing a predicate-argument representation which could not be used to classify coreferences in these structures in the detail required by the task definition. The parser also performed poorly on article headers, where the use of capitalisation for non-proper nouns introduced considerable difficulties.

The more interesting problems, from the point of view of the base algorithm, include the failure to corefer instances of classes which were simply not in the ontology. Although the results of the base algorithm variations show clearly that the dynamic creation of previously unknown semantic classes gives a considerable improvement, there are still many cases where the required subclass relations could not be determined automatically. To take one example, the failure to corefer "boss" with "chairman" was due to the lack of any node in the ontology corresponding to "boss" and the lack of any information which would allow a new node to be created as a sub- or super-class of "chairman". Techniques for extending the ontology, either by importing prexisting conceptual hierachies or by automatically, or semi-automatically, acquiring concept hierarchies from text, are currently being explored.

Other noticeable problems include the failure to corefer non-pleonastic pronouns due simply to the current distance restrictions. The definition of a class of pronouns with mandatory references, such as most personal pronouns, could then permit a mechanism to gradually extend the initial distance restriction for this class until a resolution is found. Cataphoric references, however, would still require special treatment. Lack of gender information for common forenames was also a problem.

We conclude with two general observations. First, the evaluation of coreference algorithms against a benchmark corpus is invaluable for focussing attention on phenomena which may not have been considered and for providing implicit relative frequency information about the occurrence of different classes of coreferential phenomena (for instance, coreference involving proper names is very common). Second, the LaSIE system provides an excellent base for exploring coreference algorithms by supplying a base mechanism that allows (in principle) all entities in a text to be pairwise compared for coreference, and then allows constraints to be layered on top of this base mechanism to eliminate coreferences. These constraint heuristics have access to information both supplied in the text and stored in a background world model, information both about surface forms and their position in the text and about the about compatibility of semantic classes and attributes.

Acknowledgements

None of the work reported here would have been possible without the ARPA sponsorship of MUC-6, the kind invitation of the MUC-6 organising committee for us to take part, and the contributions of many members of the MUC community who helped to formulate and refine the coreference task definition and to mark up test data: our thanks to all of these organisations and individuals. The research at Sheffield has been supported by grants from 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). The authors would like to thank Takahiro Wakao for work on the algorithms to match variants of proper names, Saliha Azzam for contributing to the analysis of the LaSIE results, and Hamish Cunningham for the code to generate MUC-6 conformant coreference markup and for countless other bits of software glue that held it all together. Pete Rodgers, Yorick Wilks and Mike Johnson provided useful comments on drafts of the paper.

References

[CHL93] N. Chinchor, L. Hirschman, and D.D. Lewis. Evaluating message understanding systems: An analysis of the third message understanding conference (muc-3). *Computational Linguistics*, 19(3):409–449, 1993.

[Gai95] R. Gaizauskas. XI: A knowledge representation language based on cross-classification and inheritance. Technical Report CS-95-24, Department of Computer Science, University of Sheffield, 1995.

[Gol90] C. F. Goldfarb. *The SGML Handbook*. Oxford University Press, 1990.

[GS96] R. Grishman and B. Sundheim. Message understanding conference - 6: A brief history. In *Proceedings of the 16th International Conference on Computational Linguistics*, Copenhagen, June 1996.

[GWH+95] R. Gaizauskas, T. Wakao, K. Humphreys, H. Cunningham, and Y. Wilks. University of sheffield: Description of the lasie system as used for muc-6. In *Proceedings of the Sixth Message Understanding Conference (MUC-6)*. Morgan Kaufmann, 1995.

[Hob78] J.R. Hobbs. Resolving pronoun references. *Lingua*, 44:311–338, 1978. Reprinted in: B.J. Grosz, K. Spark-Jones and B.L. Webber (eds,) *Readings in Natural Language Processing*, pp. 339–352, Los Altos, Ca.: Morgan Kaufmann, 1986.

[Jac92] P.S. Jacobs, editor. *Text-Based Intelligent Systems: Current Research and Practice in Information Extraction and Retrieval*. Lawrence Erlbaum, Hillsdale, NJ, 1992.

[LL94] S. Lappin and H.J. Leass. An algorithm for pronominal anaphora. *Computational Linguistics*, 20(4):535–561, December 1994.

[MUC95] Advanced Research Projects Agency. *Proceedings of the Sixth Message Understanding Conference (MUC-6)*. Morgan Kaufmann, 1995.

[Sal89] G. Salton. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley, Reading MA, 1989.

[VBA+95] M. Vilain, J. Burger, J. Aberdeen, D. Connolly, and L. Hirschman. A model-theoretic coreference scoring scheme. Technical report, Mitre Corporation, 1995.