Event Coreference for Information Extraction

Kevin Humphreys and Robert Gaizauskas and Saliha Azzam

Department of Computer Science
The University of Sheffield
Regent Court, 211 Portobello Street
Sheffield S1 4DP UK
{K.Humphreys,R.Gaizauskas,S.Azzam}@dcs.shef.ac.uk

Abstract

We propose a general approach for performing *event* coreference and for constructing complex event representations, such as those required for information extraction tasks. Our approach is based on a representation which allows a tight coupling between world or conceptual modelling and discourse modelling. The representation and the coreference mechanism are fully implemented within the LaSIE information extraction system where the mechanism is used for both object (noun phrase) and event coreference resolution. Indirect evaluation of the approach shows small, but significant benefit, for information extraction tasks.

1 Introduction

Much recent work on anaphora has concentrated on coreference between objects referred to by noun phrases or pronouns (see, e.g., Botley and McEnery (1997)). But coreference involving events, expressed via verbs or nominalised verb forms, is also common, and can play an important role in practical applications of natural language processing (NLP) systems.

One application area of increasing interest is *information extraction (IE)* (see, e.g., Cowie and Lehnert (1996)). Information extraction systems attempt to fill predefined template structures with information extracted from short natural language texts, such as newswire articles. The prototypical IE tasks are those specified in the Message Understanding Conference (MUC) evaluations (DARPA, 1995; Grishman and Sundheim, 1996). In these exercises the main template filling task centres around a 'scenario' which is defined in terms of a key event type and various roles pertaining to it. Examples of scenarios used in previous MUCs include joint venture announcements, microprocessor product announcements, terrorist attacks, labour negotiations, and management succession events. In order not to spuriously overgenerate event instances and to properly acquire all available role information, it is crucial that multiple references to the same event be correctly identified and merged. While these concerns are of central importance to IE systems, they are clearly of significance for any NLP system, and more broadly for any computational model of natural language.

A few concrete examples will make the issues clearer[1]. A management succession event (as used in MUC-6) may involve the two separate events of a corporate position being vacated by one person and then filled by another. For an event to be considered reportable for the IE task, the post, the company and at least one person (either incoming or outgoing) must all be identifiable in the text.

The first thing to note here is that while management succession events are sometimes reported as single, simple events, as in

(1) *Mr. Jones succeeds M. James Bird, 50, as president of Wholistic Therapy.*

more frequently multiple aspects or sub-events of a single succession event are identified in separate clauses by separate verb phrases or nominalised forms:

(2) *Daniel Wood was named president and chief executive officer of EFC Records Group, a unit of London's Spear EFC PLC. He succeeds Charles Paulson, who was recently made chairman and chief executive officer of EFC Records Group North America.*

---

[1] All examples in this paper are taken from the MUC-6 *Wall Street Journal* corpus with names of individuals and companies changed.

75

(3) *The sell-off followed the resignation late Monday of Freddie Heller, the president of Renard Broadcasting Co. Yesterday, Renard named Susan B. Kempham, chairman of Renard Inc.'s television production arm, to succeed him.*

Both of these pairs of sentences refer to a single management succession event (though the second sentence in 2 also identifies a further one). Such event/sub-event relations are similar to the familiar part-whole or related-object anaphora exemplified in sentences such as *The airplane crashed after the wings fell off* or *When John entered the kitchen the stove was on* (Allen, 1987).

The second thing to note is the variety of surface forms used to refer to events. Events are referred to by verb phrases in main clauses (1 above), and in relative clauses (second sentence in 2) or subordinate clauses. They may be referred to through nominalised forms (*resignation* in 3 above) or through infinitival forms in control sentences (second sentence in 3). When there are multiple references to the same event, antecedent and anaphor appear to be able to adopt all combinations of these forms[2].

This paper discusses an approach to handling event coreference as implemented in the LaSIE information extraction system (Gaizauskas et al., 1995; Gaizauskas and Humphreys, 1997b). Within this system, event coreference is handled as a natural extension to object coreference, outlined here and described in detail in Gaizauskas and Humphreys (1997a). Both mechanisms are handled within a general approach to discourse and world modelling.

In the next section we give a brief overview of the LaSIE system. Section 3 describes in more detail the approach to world and discourse modelling within LaSIE and Section 4 details our coreference procedure. In Section 5 we discuss a particular example in detail and show how our approach enables us to correctly corefer multiple event references. Section 6 presents results of an approach to evaluating the the approach and Section 7 concludes the paper with some general discussion.

2 LaSIE Overview

The Large Scale Information Extraction system (LaSIE) has been designed as a general purpose IE research system, initially geared towards, but not

---

[2]While no extended study has been carried out, it appears that in newswire texts nominalised forms are less likely to appear in the first reference to an event, and more likely to appear in subsequent references.

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 DARPA (1995) for further details of the task descriptions). In addition, the system can generate a brief natural language summary of any scenario it has detected in a text. All these tasks are carried out by building a single rich discourse model of the text from which the various results are read off. The system is a pipelined architecture which processes a text one 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 (see Gaizauskas et al. (1995) for further details):

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;

parsing and semantic interpretation builds lexical and phrasal chart edges in a feature-based formalism then 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', constructs a predicate-argument representation of the current sentence;

discourse interpretation adds the information from the predicate-argument representation to a hierarchically structured semantic net which encodes the system's world model, adds additional information presupposed by the input, performs coreference resolution between new and existing instances in the world model, and adds any information consequent upon the new input.

2.1 MUC-6 Coreference Performance

MUC-6 included a quantitatively evaluated coreference task, which required participating systems to propose coreference annotations for a set of texts. These annotations were then automatically scored against manually produced annotations for the same texts. The performance of the LaSIE system in this coreference task was 51% recall and 71% precision. This compares favourably with the highest scoring MUC-6 systems: the highest recall system scored 63% recall and 63% precision; the highest precision system scored 59% recall and 72% precision. Recall

here is a measure of how many correct (i.e. manually annotated) coreferences the system actually found, and precision is a measure of how many coreferences the system proposed were actually correct. For example, suppose there are 100 real coreference relations in a corpus and a system proposes 75, of which 50 are correct. Then its recall is 50/100 or 50% and its precision is 50/75 or 66.7%.

The MUC-6 definition of the coreference task included several forms of NP coreference, not only pronominal relations. However, it did not include event coreference, which can be measured only indirectly via the information extraction task results, a topic to which we return in Section 6.

3 Discourse Interpretation

The LaSIE system's 'world' or domain of interest is modelled by an inheritance-based semantic graph, using the XI knowledge representation language (Gaizauskas, 1995). In the graph classes of objects, events, and attributes appear as nodes; each node may have associated with it an attribute-value structure and these structures are inherited down the graph. The higher levels of the graph, or *ontology*, for the management succession task have the structure shown in Figure 1. Two simple attribute-value structures are also shown in the graph, connected by dashed lines to the nodes with which they are associated.

Attribute-value structures are just sets of attribute:value pairs where the value for an attribute may either be static, as in the pair animate:yes, which is associated with the person node, or dynamic, where the value is dependent on various conditions, the evaluation of which makes reference to other information in the model. Certain special attribute types, presupposition and consequence, may also return values which are used at specific points to modify the current state of the model.

As a discourse is processed, discourse entities (objects and events introduced by the text) are added as new nodes in the graph beneath their parent class and have associated with them an attribute-value structure containing both inherited and discourse-supplied attributes. This process may involve hypothesising new implicit entities if they are not available explicitly in the text, or have not been discovered by the parser, but are required role players for a given event type. Knowledge about required roles is represented via attributes in the world model. For example, in Figure 1 we see that a retire event requires a logical subject of type

person and an entity of this type will be hypothesised if it is not available from the text.

4 Coreference Resolution

After each sentence in a text is added to the 'world model', gradually forming a discourse-specific model, a coreference procedure is applied to attempt to resolve, or merge, each of the newly added instances with instances currently in the discourse model. Coreference resolution is performed by comparing instances from several candidate sets, each of which is a set of pairs of instances where one element is an instance from the current input sentence and the other an instance occurring earlier in the text, which may be coreferential. The algorithm proceeds as follows for each instance pair being considered:

1. Ensure semantic type consistency

   To determine semantic consistency requires establishing a path in the semantic graph between the semantic types of the two instances. If a path can be found a semantic similarity score is calculated using the inverse of the length of the path (measured in nodes) between the two types.

   For event instances, a path is valid if both event types are dominated by a task-specific top node, i.e. both types must be potential sub-events of an event required by the current IE template. For example, 'hire' and 'retire' are both sub-events of the 'succession' event in the ontology sketched above.

   For instances of the object class, a path is valid if the two types stand in a dominance relation, i.e. the types are ordered in the ontology. For example, 'company' is a sub-class of 'organisation' so these type are ordered (and have a semantic similarity score of 0.5).

   If no valid path can be found the attempt to resolve the two instances is abandoned.

2. Ensure attribute consistency

   Certain attributes, e.g. animate and time, are specified in the ontology as taking a single fixed value for any particular instance. If two instances being compared have a common attribute of this type, the values must be identical or the attempted resolution is abandoned.

   Type specific coreference constraints are then examined by attempting to inherit a distinct attribute. If this attribute can be derived from any of the instances' superclasses the attempted resolution of the current pair is abandoned.

77

entity

object event attribute

person organisation date succession single-valued multi-valued

company government incoming outgoing animate count name near
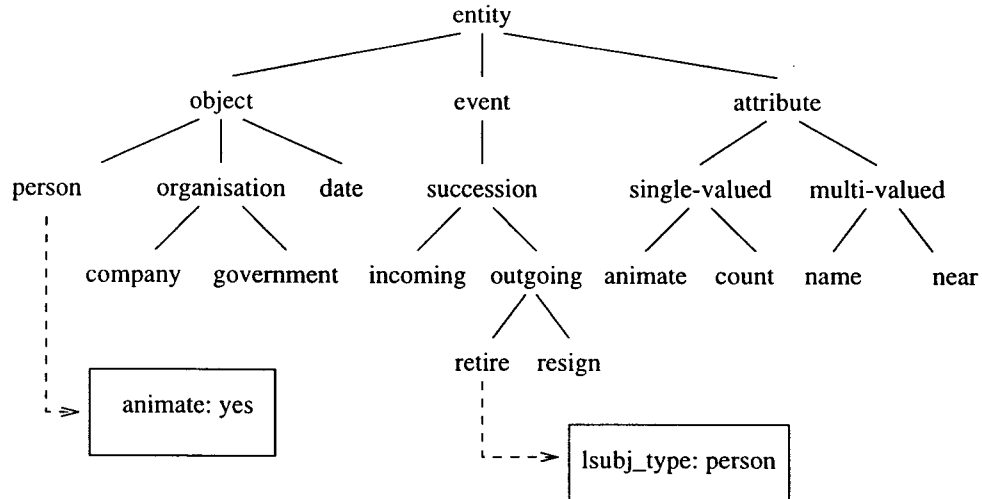
retire resign

animate: yes

lsubj_type: person

Figure 1: Upper ontology for the management succession task

Constraints on the various event types are detailed in the following section.

3. 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.

After each pair in a candidate 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 will be merged. 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.

4.1 Event Coreference

The constraints on events as used in Step 2 of the general coreference algorithm above can be associated with any event node in the ontology, and will then be inherited by all instances of all sub-event types. The constraints currently used can be categorised in the following way:

1. General task-independent constraints are associated with the top-level event node. For example two event instances are defined as

distinct (i.e. not coreferential) if they have incompatible times. At present this simply means that two events with different tenses cannot be resolved, but clearly a more detailed model of event times is required, particularly as Crowe (1996) shows how temporal phrases are consistently useful in distinguishing and recognising events[3].

2. General task-specific constraints are, for the management succession task, associated with the succession_event node. For example, the constraint that two instances must be distinct if they involve different organisations or different management positions.

3. More specific constraints are represented at lower and possibly verb-specific nodes. For example, an incoming_event (e.g. hire, promote) is distinct from a changeover_event (e.g. replace, succeed) if the former's logical object is distinct from the latter's logical subject.

The determination of distinct or compatible event roles requires the application of the coreference mechanism to instances of the object class (the role players in the event). The same algorithm is used but the inherited constraints will be those associated with the object nodes in the ontology. For ex-

[3]It is possible to represent a time scale within the current XI formalism and then associate each input event with a point on the scale. Each point can be treated as a potential interval and be expanded to include the times of sub-events. The representation and use of this more detailed model is currently under investigation.

ample, indefinite noun phrases cannot be anaphors, pronouns should be resolved within the current paragraph, definite noun phrases within the last two paragraphs, etc. Full details and an evaluation of the coreference constraints on object instances can be found in Gaizauskas and Humphreys (1997a).

The constraints above are similar to those used in the FASTUS IE system (Appelt et al., 1995) and by Sowa (1984), where the merging takes place between template structures, considering special conditions for the unification of variables in template slots. However, the general approach here has more in common with Whittemore and Macpherson (1991) or Zarri (1992), where event merging is carried out within the underlying knowledge representation.

5 A Worked Example

This section describes the operation of the general coreference mechanism for the example (3) presented in the introduction, concentrating on the effect of the various constraints on event instances. We reproduce the two sentences in (3) here:

(3a) *The sell-off followed the resignation late Monday of Freddie Heller, the president of Renard Broadcasting Co.*

(3b) *Yesterday, Renard named Susan B. Kempham, chairman of Renard Inc.'s television production arm, to succeed him to succeed him.*

The full semantic representation of these sentences as produced by the parser/semantic interpreter for input to the discourse interpreter is:

```
% Sentence 3a
sell-off(e2), number(e2,sing), det(e2,the),
follow(e1), time(e1,past),
   lsubj(e1,e2), lobj(e1,e3),
resignation(e3), number(e3,sing),
   det(e3,the),
date(e5), name(e5,'Monday'),
person(e7), name(e7,'Freddie Heller'),
title(e8,president),
company(e10),
   name(e10,'Renard Broadcasting Co.')


% Sentence 3b
yesterday(e11), number(e11,sing),
name(e13,'Renard'),
name(e12), time(e12,past),
   lsubj(e12,e13),
person(e15), name(e15,'Susan B. Kempham'),
   apposed(e15,e16),              ·
title(e16,chairman),
```

```
arm(e18), number(e18,sing),
   qual(e18,e19),
production(e19), number(e19,sing),
   qual(e19,e20), of(e19,e21),
television(e20),
company(e21), name(e21,'Renard Inc.'),
succeed(e14), time(e14,present),
   lsubj(e14,e15), lobj(e14,e22),
pronoun(e22,him)
```

The nominalisation of the verb *resign* in (3a) leads to the presupposition of an outgoing_event, which in turn leads to hypothesised objects for a related person, post and organisation (these presuppositions are stored as attributes of the outgoing_event in the world model). The coreference mechanism will then be applied to these objects and, in this case, will be able to resolve all three within the same sentence. The resign event therefore forms a complete succession event for the management succession IE task.

Both verbs in (3b), the incoming_event *name* and the changeover_event *succeed*, will cause the introduction of succession event instances into the discourse model, each of which will cause the hypothesis of a related person, post and organisation. Attributes of the name event will add additional constraints to its hypothesised objects, including the specification that the organisation should be a potential subject of the verb, the person a potential logical object, and the post a potential complement. Objects with the required features will be found by the coreference mechanism for the organisation and person, but not the post. The succeed event will also cause the hypothesis of an additional person, with the constraints that one must be incoming, and a potential logical subject of the verb, and the other outgoing, and a potential logical object. The succeed event's hypothesised organisation and post will be resolved with the same objects as the resign event from the previous sentence.

The general constraints on coreferential succession events are therefore satisfied for the succeed and resign events, and the restrictions on the more specific subclasses must then be considered. The relevant restriction here is that a changeover_event must share its logical object with the logical subject of an outgoing_event. This will require the application of the coreference mechanism for objects to resolve the pronoun *him*. A correct resolution with *Freddie Heller* will then allow the two events to be resolved.

The succeed and name events will also be resolved similarly, using the restriction that a changeover_event must share its logical subject

79

	Succession Events		Overall		
	Recall	Precision	Recall	Precision	Combined
No Event Coref	66%	72%	42%	59%	48.88%
With Event Coref	65%	77%	42%	60%	49.40%

Table 1: Upper Ontology for the management succession task

with the logical object of an incoming_event. In this case the infinitive form of the *succeed* verb will have no explicit logical subject, but one will be hypothesised and resolved with the best antecedent of the required type (person), here *Susan B. Kempham*. The two events can therefore be merged, to result in the representation of a single succession event with *Freddie Heller* outgoing and *Susan B. Kempham* incoming.

6 Evaluation

We have not been able to carry out direct evaluation of our approach to event coreference. To do so would require manually annotating coreferential events in a corpus of significant size, and we have not had the resources to do so. However, we have attempted to gain some indirect measure of the successfulness of the approach by toggling event coreference on and off and observing the effect on the ability of the system to fill MUC-6 management succession templates correctly. The hypothesis here is that effective event coreference will lead to higher scores in the template filling task for at least two reasons. First, role players in events (which become slot fillers in the scored templates, e.g. persons and organisations) should become available due to event coreference. Second, spurious succession events should be eliminated due to proper event coreference.

The MUC-6 management succession scenario task involved filling an object-oriented template consisting of five objects, each with associated slots (twenty slots in total). The top level object was a template object and contained one or more succession_event objects which in turn contained an organization object and one or more in_and_out objects, themselves containing organization and person objects (a precise definition of the template and the task can be found in DARPA (1995)).

Table 1 shows the gross results of running the system against the 100 articles in the MUC-6 scenario task test corpus. Our system is easily reconfigured to run with or without attempting event coreference. The two rows in the table show the effects without and with event coreference. The 'Overall'

column show the effects on the overall scenario template filling task, i.e., on recall and precision scores for all objects and slots in the templates. The 'Succession Events' column shows the effect just for the succession_event objects in the templates, and is therefore a more direct measure of template filling performance where we might expect event coreference to have an effect.

As can be seen from the table, the effect overall is not particularly significant. However, the effect on succession events alone is more substantial, with precision going up five percentage points and recall dropping only one, when event coreference is switched on. Closer examination revealed that the event coreference mechanism successfully avoided the proposal of 11 spurious succession events in the evaluation corpus, which included 196 possible events.

We stress that this is a crude measure of our event coreference algorithm – really just an indication of its utility in the information extraction task. However, even as such, it shows that the algorithm is performing correctly, on balance, and that event coreference is worth addressing in an IE system.

7 Conclusion

Event coreference is more complex than object coreference because of the requirement that objects filling particular event roles in two possibly coreferential events must themselves be coreferential. Coreferring events is therefore logically secondary to coreferring objects[4].

The approach we describe here provides a very general and powerful mechanism for performing event coreference and for constructing complex event representations, such as those required for information extraction tasks. Within information extraction the problem has typically been addressed by attempting to merge, or unify, extracted templates (e.g. Sowa (1984) or Appelt et al. (1995)), but a more generally useful

---

[4]Of course in some events, roles may be filled by other events, but this complication does not affect the basic point that object coreference is primary and event coreference dependent upon it.

mechanism will operate within a more general representation. Our approach can be compared to that of Whittemore and Macpherson (1991) who discuss incremental building of event representations within a modified form of DRT (Kamp, 1981). However, the representation used here is preferred because it allows a tighter coupling between world or conceptual modelling and discourse modelling.

The representation and the coreference mechanism are fully implemented within the LaSIE information extraction system and are currently being extended to make use of a richer model of event times, the importance of which is demonstrated in Crowe (1996). The mechanism described here is used in the LaSIE system for both object and event coreference, treating the different types simply as instances subject to differing constraints, where constraints can be easily represented at any level of generality. Our evaluation, while far from exhaustive, shows that addressing event coreference can clearly result in real benefits for IE systems.

8 Acknowledgements

References

Allen, J. 1987. *Natural Language Understanding.* Benjamin/Cummings, Menlo Park, CA, 1st edition.

Appelt, D., J. Hobbs, J. Bear, D. Israel, M. Kameyama, A. Kehler, D. Martin, K. Myers, and M. Tyson. 1995. SRI International FASTUS system: MUC-6 Test Results and Analysis. In *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, pages 237–248. Morgan Kaufmann.

Botley, S. and T. McEnery, editors. 1997. *Discourse Anaphora and Anaphor Resolution.* University College London Press, London. In press.

Cowie, J. and W. Lehnert. 1996. Information Extraction. *Communications of the ACM*, 39(1):80–91.

Crowe, J. 1996. *Constraint Based Event Recognition for Information Extraction.* Ph.D. thesis, University of Edinburgh.

DARPA. 1995. *Proceedings of the Sixth Message Understanding Conference (MUC-6).* Morgan Kaufmann.

Gaizauskas, R. 1995. XI: A Knowledge Representation Language Based on Cross-Classification and Inheritance. Technical Report CS-95-24, Department of Computer Science, University of Sheffield.

Gaizauskas, R. and K. Humphreys. 1997a. Quantative Evaluation of Coreference Algorithms in an Information Extraction System. In S. Botley and T. McEnery, editors, *Discourse Anaphora and Anaphor Resolution.* University College London Press. In press.

Gaizauskas, R. and K. Humphreys. 1997b. Using a Semantic Network for Information Extraction. *Journal of Natural Language Engineering.* In press.

Gaizauskas, R., T. Wakao, K Humphreys, H. Cunningham, and Y. Wilks. 1995. Description of the LaSIE System as Used for MUC-6. In *Proceedings of the Sixth Message Understanding Conference (MUC-6).* Morgan Kaufmann.

Grishman, R. and B. Sundheim. 1996. Message Understanding Conference - 6: A Brief History. In *Proceedings of the 16th International Conference on Computational Linguistics*, Copenhagen, June.

Kamp, H. 1981. A Theory of Truth and Semantic Representation. In *Formal Methods in The Study of Language.* J. Groenendijk, Jannsen, T, and Stokhof, M.

Sowa, J.F. 1984. *Conceptual Structures : Information Processing in Mind and Machine.* Reading (MA): Addison-Wesley.

Whittemore, S and M. Macpherson. 1991. Event-building through Role-Filling and Anaphora Resolution. In *Proceedings of the 29th meeting of the Asssociation for Computational Linguistics*, Berkeley, CA.

Zarri, G.P. 1992. Semantic Modeling of the Content of (Normative) Natural Language Documents. In *Proceedings of the specialized conference on Natural Language Processing and its applications*, Avignon.