CONSTRAINT-BASED EVENT RECOGNITION FOR INFORMATION EXTRACTION

Jeremy Crowe*
Department of Artificial Intelligence
Edinburgh University
Edinburgh, EH1 1HN
UK
j.crowe@ed.ac.uk

Abstract

We present a program for segmenting texts according to the separate events they describe. A modular architecture is described that allows us to examine the contributions made by particular aspects of natural language to event structuring. This is applied in the context of terrorist news articles, and a technique is suggested for evaluating the resulting segmentations. We also examine the usefulness of various heuristics in forming these segmentations.

Introduction

One of the issues to emerge from recent evaluations of information extraction systems (Sundheim, 1992) is the importance of discourse processing (Iwańska et al., 1991) and, in particular, the ability to recognise multiple events in a text. It is this task that we address here.

We are developing a program that assigns message-level event structures to newswire texts. Although the need to recognise events has been widely acknowledged, most approaches to information extraction (IE) perform this task either as a part of template merging late in the IE process (Grishman and Sterling, 1993) or, in a few cases, as an integral part of some deeper reasoning mechanism (e.g. (Hobbs et al., 1991)).

Our approach is based on the assumption that discourse processing should be done early in the information extraction process. This is by no means a new idea. The arguments in favour of an early discourse segmentation are well known – easier coreference of entities, a reduced volume of text to be subjected to necessarily deeper analysis, and so on.

Because of this early position in the IE process, an event recognition program is faced with a necessarily shallow textual representation. The purpose of our work is, therefore, to investigate the quality of text segmentation that is possible given such a surface form.

Event recognition

What is an event?

If we are to distinguish between events, it is important that we know what they look like. This is harder than it might at first seem. A closely related (though not identical) problem is found in recognising boundaries in discourse, and there seems to be little agreement in the literature as to the properties and functions they possess (Morris and Hirst, 1991), (Grosz and Sidner, 1986).

Our system is aimed at documents typified by those in the MUC-4 corpus (Sundheim, 1992). These deal with Latin American terrorist incidents, and vary widely in terms of origin, medium and purpose. In the task description for the MUC-4 evaluation, two events are deemed to be distinct if they describe either multiple types of incident or multiple instances of a particular type of incident, where instances are distinguished by having different locations, dates, categories or perpetrators. (NRaD, 1992)

Although this definition suffers from a certain amount of circularity, it nonetheless points to an interesting feature of events at least in so far as physical incidents are concerned. It is generally the case that such incidents *do* possess only one location, date, category or description. Perhaps we can make use of this information in assigning an event-segmentation to a text?

Current approaches

As an IE system processes a document, it typically creates a template for each sentence (Hobbs, 1993), a frame-like data structure that contains a maximally explicit and regularised representation of the information the system is designed to extract. Templates are merged with earlier ones unless they contain incompatible slot-fills.

Although more exotic forms of event recognition exist at varying levels of analysis (such as within the abductive reasoning mechanism of SRI's TACITUS system (Hobbs et al., 1991), in a thesaurus-based lexical cohesion algorithm (Morris and Hirst, 1991) and in a semantic network (Kozima, 1993)), template merging is the most used method.

*I would like to thank Chris Mellish and the anonymous referees for their helpful comments. Supported by a grant from the ESRC.

Modular constraint-based event recognition

The system described here consists of (currently) three *analysis modules* and an *event manager* (see figure 1). Two of the analysis modules perform a certain amount of island-driven parsing (one extracts time-related information, and the other location-related information), and the third is simply a pattern matcher. They are designed to run in parallel on the same text.

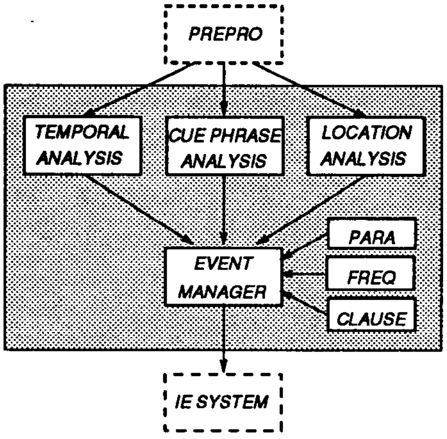


Figure 1: System architecture

Analysis modules

The fragments of natural language that represent time and location are by no means trivial to recognise, let alone interpret. Consequently, and in keeping with the fast and shallow approach we have adopted, the range of spatio-temporal concepts the program handles has been restricted.

For example, the semantic components of both modules know about points in time/space only, and not about durations. There are practical and theoretical reasons for this policy decision – the aim of the system is only to distinguish between events, and though the ability to represent durations is in a very few situations useful for this task, the engineering overheads in incorporating a more complex reasoning mechanism make it difficult to do so within such a shallow paradigm.

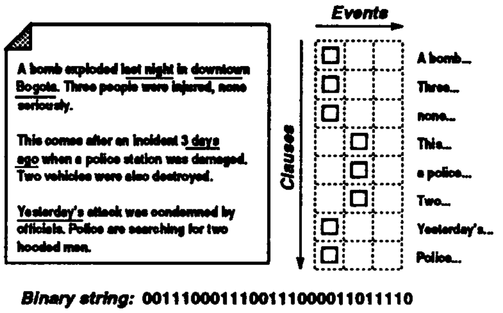
The first two analysis modules independently assign explicit, regularised PATR-like representations to the time- and location-phrases they find. Graph unification is then used to build a set of constraints determining which clauses[1] in a text can refer to the same event. Each module then passes its constraints to the event manager.

The third module identifies sentences containing a subset of cue phrases. The presence of a cue phrase in a sentence is used to signal the start of a (totally) new event.

Event manager

The role of the event manager is to propose an event segmentation of the text. To do this, it makes use of the constraints it receives from the analysis modules combined with a number of document-structuring heuristics. Many clauses ("quiet clauses") are free from constraint relationships, and it is in these cases that the heuristics are used to determine how clauses should be clustered.

A text segmentation can be represented as a grid with clauses down one side, and events along the other. Figure 2 contains a representation of a sample news text, and shows how this maps onto a clause/event grid. The phrases overtly referring to time and location have been underlined.


A bomb exploded last night in downtown Bogota. Three people were injured, none seriously.

This comes after an incident 3 days ago when a police station was damaged. Two vehicles were also destroyed.

Yesterday's attack was condemned by officials. Police are searching for two hooded men.

Figure 2: Example text segmentation

Structuring strategies

Although the legal event assignments for a particular clause may be restricted by constraints, there may still be multiple events to which that clause *can* be assigned.

Three structuring strategies are being investigated. The first dictates that clauses should be assigned to the lowest non-conflicting event value; the second favours non-conflicting event values of the most recently assigned clauses. The third strategy involves a mix of the above, favouring the event value of the previous clause, followed by the lowest non-conflicting event values.

Heuristics

Various heuristics are used to gel together quiet clauses in the document. The first heuristic operates at the paragraph level. If a sentence-initial clause appears in a sentence that is not paragraph-initial, then it is assigned to the same event as the first clause in the previous sentence. We are therefore making some assumptions about the way reporters structure their articles, and part of our work will be to see whether such assumptions are valid ones.

The second heuristic operates in much the same way as the first, but at the level of sentences. It is based on the reasoning that quiet clauses should be assigned to the same event as previous clauses within the sentence. As such, it only operates on clauses that are not sentence-initial.

Finally, a third heuristic is used which identifies similarities between sentences based on n-gram frequencies (Salton and Buckley, 1992). Areas to investigate are the optimum value for n, the effect of normalization

[1] A clause in this case is delimited in much the same way as in Hobbs et al's terminal substring parser (Hobbs et al., 1991), i.e. by commas, relative pronouns, some conjunctions and some forms of `that`.

on term vector calculation, and the potential advantages of using a threshold.

This heuristic also interacts with the text structuring strategies described above; when it is activated, it can be used to override the default strategy.

Experiments and evaluation

Whilst the issue of evaluation of information extraction in general has been well addressed, the evaluation of event recognition in particular has not. We have devised a method of evaluating segmentation grids that seems to closely match our intuitions about the "goodness" of a grid when compared to a model.

The system is being tested on a corpus of 400 messages (average length 350 words). Each message is processed by the system in each of 192 different configurations (i.e. with/without paragraph heuristic, varying the clustering strategy etc.), and the resulting grids are converted into binary strings. Essentially, each clause is compared asymmetrically with each other, with a "1" denoting a difference in events, and a "0" denoting same events.

Figure 2 shows an example of a binary string corresponding to the grid in the same figure. Figure 3 shows a particular 4-clause grid scored against all other possible 4-clause grids, where the grid at the top is the intended correct one, and the scores reflect degrees of similarity between relevant binary strings.

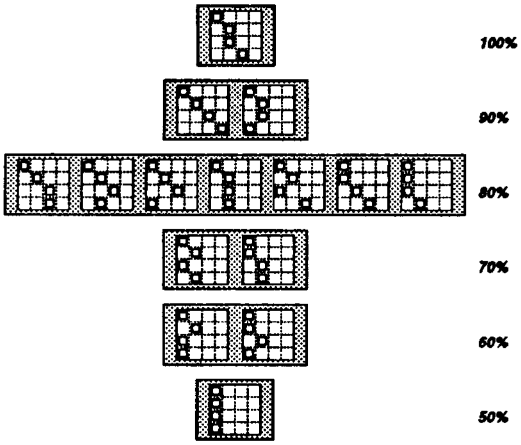


Figure 3: Comparison of scores for a 4-clause grid

In order to evaluate these computer generated grids, a set of manually derived grids is needed. For the final evaluation, these will be supplied by naïve subjects so as to minimise the possibility of any knowledge of the program's techniques influencing the manual segmentation.

Conclusions and future work

We have manually segmented 100 texts and have compared them against computer-generated grids. Scoring has yielded some interesting results, as well as suggesting further areas to investigate.

The results show that fragments of time-oriented language play an important role in signalling shifts in event structure. Less important is location information – in fact, the use of such information actually results in a slight overall *degradation* of system performance. Whether this is because of problems in some aspect of the location analysis module, or simply a result of the way we use location descriptions, is an area currently under investigation.

The paragraph and clause heuristics also seem to be useful, with the omission of the clause heuristic causing a considerable degradation in performance. The contributions of *n*-gram frequencies and the cue phrase analysis module are yet to be fully evaluated, although early results are encouraging.

It therefore seems that, despite both the shallow level of analysis required to have been performed (the program doesn't know what the events actually *are*) and our simplification of the nature of events (*we* don't know what they really are either), a modular constraint-based event recognition system is a useful tool for exploring the use of particular aspects of language in structuring multiple events, and for studying the applicability of these aspects for automatic event recognition.

References

Ralph Grishman and John Sterling. 1993. Description of the Proteus system as used for MUC-5. In *Proc. MUC-5*. ARPA, Morgan Kaufmann.

Barbara Grosz and Candy Sidner. 1986. Attention, intensions and the structure of discourse. *Computational Linguistics*, 12(3).

Jerry R Hobbs, Douglas E Appelt, John S Bear, Mabry Tyson, and David Magerman. 1991. The TACITUS system. Technical Report 511, SRI.

Jerry R Hobbs. 1993. The generic information extraction system. In *Proc. MUC-5*. ARPA, Morgan Kaufmann.

Lucja Iwańska, Douglas Appelt, Damaris Ayuso, Kathy Dahlgren, Bonnie Glover Stalls, Ralph Grishman, George Krupka, Christine Montgomery, and Ellen Riloff. 1991. Computational aspects of discourse in the context of MUC-3. In *Proc. MUC-3*, pages 256–282. DARPA, Morgan Kaufmann.

Hideki Kozima. 1993. Text segmentation based on similarity between words. In *Proc. ACL, student session*.

Jane Morris and Graeme Hirst. 1991. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, 17(1):21–42.

NRaD. 1992. MUC-4 task documentation. NRaD (previously Naval Ocean Systems Center) On-line document.

Gerald Salton and Chris Buckley. 1992. Automatic text structuring experiments. In Paul S Jacobs, editor, *Text-Based Intelligent Systems*, chapter 10, pages 199–210. Lawrence Erlbaum Associates.

Beth M Sundheim. 1992. Overview of the fourth message understanding conference. In *Proc. MUC-4*, pages 3–21. DARPA, Morgan Kaufmann.