On the Importance of Annotating Event-Event Temporal Relations in Text

Andrea Setzer and Robert Gaizauskas

Department of Computer Science
University of Sheffield
Regent Court
211 Portobello Street
Sheffield S1 4DP, UK
{A.Setzer, R.Gaizauskas}@dcs.shef.ac.uk

Abstract

Many natural language processing applications, such as information extraction, question answering, topic detection and tracking, and multi-document summarisation, would benefit significantly from the ability to accurately position reported events in time, either relatively with respect to other events or absolutely with respect to calendrical time. However, only recently has concerted work started on the automatic extraction of temporal information from text. The overall aim of our work is to automatically establish the temporal relations holding between events as well as between events and calendrical times in newspaper articles. This information makes it possible to create a 'time-event graph' to represent the temporal information contained in a text, and would in turn support the applications mentioned above. In this paper we first argue for the superiority of the time-event graph over a time-stamped event sequence as a target representation for extracted temporal information and discuss the importance of annotating temporal relations. We then give a brief account of the annotation scheme we have devised which allows us to annotate relational information as well as temporal referring expressions. We also discuss a pilot study in which we assessed the utility and feasibility of the scheme and the annotation tool we have developed to aid the annotation process. Finally, we discuss potential improvements in the annotation tool which are aimed at making the annotation of larger scale corpora possible.

1. Introduction

Many natural language processing applications, such as information extraction, question answering, topic detection and tracking, and multi-document summarisation, would benefit significantly from the ability to accurately position reported events in time, either relatively with respect to other events or absolutely with respect to calendrical time. However, only recently has concerted work been started on the automatic extraction of temporal information from text.

In addressing the goal of extracting temporal information from text, it is necessary to:

1. specify the target temporal representation which we wish to obtain for a text;
2. identify ancillary information which we **may** want to extract because of its utility in arriving at the target temporal representation (by analogy with, e.g. part-of-speech tagging or parsing as intermediate goals towards semantic interpretation).

For example, one candidate for target representation is an association of a calendrical time point or interval with each event in a text, i.e. a list of pairs of calendrical times and events. Arriving at this representation might require extracting additional information, such as temporal relational information, about events. For example, assigning "before 1984" to an event A might only be possible by recognising that event B occurs in 1984 and that A occurs before B. Thus, the capability to determine temporal relations between events might be a useful component capability in a temporal information extraction system, even if the information identified by such a component is not directly included in the target representation.

Our view is that target representation should be a **time-event graph** where the nodes in the graph are either times or events and the arcs are temporal relations. This is somewhat different from the "time-stamping" representation introduced in the preceding paragraph and one of our major goals in this paper is to argue that it is a superior representation.

With respect to ancillary temporal information to be extracted, our view is that time-referring expressions, event representatives, and temporal relations as signalled by, e.g. prepositions and temporal adverbials, all convey important temporal information and should be extracted. This information is necessary to derive a time-event graph for a text; but of course it is useful for creating a time-stamp representation as well – arguably both necessary and sufficient.

In this paper, we first give an overview over existing approaches to temporal annotation and information extraction in Section 2. Then in Section 3. we discuss the importance of a target representation that captures temporal relations and describe the annotation scheme we have developed to do so. Section 4. presents some results of a pilot study we have conducted based on the scheme. Further improvements to the process of annotation, to support the creation of larger annotated resources, are discussed in section 5.

2. Overview of Existing Approaches

Existing approaches to capturing temporal information in text can be divided broadly into the following three groups: (1) approaches that concentrate on an accurate and detailed annotation of temporal referring expressions, (2) time-stamping approaches that aim to associate a calendrical time with some or all events in the text, and (3) approaches that focus on the temporal relations between events and times, between events and events or both. We give a brief overview of existing work on each approach in this section.

2.1. Annotating Temporal Referring Expressions

The most extensive work on annotating temporal referring expressions so far has been done as part of the MUC language technology evaluations or the subsequent TIDES[1] and ACE [2] programmes.

2.1.1. MUC Named Entity Task

Between 1987 and 1998 the DARPA-sponsored Message Understanding Conferences (MUCs) developed a quantitative evaluation regime for message understanding (MU) systems, now generally called information extraction (IE) systems. The last MUC, MUC-7, was held in 1998, but related work continues within the ACE workshops. For more information about the message understanding conferences see MUC (1998).

While MUC evaluations typically defined several evaluation tasks, the relevant task here is the *named entity (NE) recognition* task, introduced in MUC-5. The NE task required the recognition and classification of specified named entities such as persons, locations, organisations, monetary amounts and, most importantly in the current context, time expressions (timex). The aim of the timex task was to mark up time expressions in text using SGML tags and to classify these expressions using a TYPE attribute. Type DATE referred to complete or partial date expressions and type TIME referred to complete or partial expressions of time of day. Both absolute and relative time expressions had to be marked up, although these two types were not distinguished in the annotation.

In the MUC-7 evaluation, the best systems were able to obtain F-measure scores approaching 94% on this task.

2.1.2. An Annotation Scheme for Temporal Expressions

Wilson et al. (2001) describe a set of guidelines[3] being developed within the TIDES programme for annotating time expressions and associating with them a canonical representation of the times to which they refer. A method for extracting such time expressions from multiple languages is also introduced. The main novel features as compared to the MUC temporal annotation task are:

1. In MUC the task called merely for surface time expressions to be annotated and crudely classified, whereas the Wilson et al. (2001) guidelines also call for each expression to be *evaluated*, i.e., to have associated with it a normalised representation of the time referred to.

2. The range of expressions flagged is much wider.

3. Context-dependent time expressions like *today* are handled in addition to fully specified time expressions like *September 3rd, 1997*. Context can be local (within the same sentence) or global (outside the sentence). Indexical time expressions, that require knowledge about the time of speech, like *now* are also included. A corpus study (Wilson and Mani, 2000) showed that two-thirds of time expressions in print and broadcast news are context dependent, so this feature is significant.

Wilson et al. (2001) have developed a tagger to do time expression tagging as described in the TIDES guidelines, and report F-measure scores of 96.2% on expression identification and 83.2% on evaluating these expressions.

2.2. Time-Stamping of Events

Annotating temporal referring expressions is only a first step towards extracting rich temporal information from text. The approaches introduced in this section aim at 'stamping' some or all events in a text with a calendar time – possibly the time value of an associated temporal referring expression.

2.2.1. MUC-5 and MUC-7 Time Slots

In addition to the Named Entity time expression tagging task, MUC-5 and MUC-7 also required relations between times and events to be established as part of the scenario template task. Participants were required to assign a calendrical time to certain specified event types (joint venture announcements and rocket launchings, respectively).

Scenario template filling requires the identification of specific relations holding between template elements. For example, the MUC-7 scenario template filling task concerned rocket launch events. The scenario template contains information about vehicles, pay load, launch site, mission function etc. It also contained a slot called LAUNCH_DATE, which was to be filled with a link to a time entity which in turn contained slots for a normalised representation of the start and end times of the temporal interval containing the launch event, if the interval could be determined from the text.

Temporal relations between events and other events were not explicitly addressed, though insofar as they were necessary to infer correct slot fills, systems needed to take them into account. Scores were quite low on this slot reflecting the difficulty of correctly assigning to it.

2.2.2. Assigning Time-Stamps to Event Clauses

In the MUC task, times were only to be determined for the events of interest, the scenario events. A more ambitious goal is to attempt to associate calendrical times or time intervals with *every* event in a text.

Filatova and Hovy (2001) describe a method for breaking news stories into their constituent events and assigning time-stamps to them. The time-stamps assigned are either full specified calendrical dates, sets of dates, closed date ranges (both end points specified), or date ranges open at one end or the other, indicating some time before or after the specified date.

The syntactic units conveying events are assumed to be simple clauses and they are identified using a parser which produces semantically labelled syntactic parse trees. Some problems are ignored in this approach, for example multiple verbs with different tenses in one sentence.

The time-stamper uses two time-points for anchoring. One time-point is the time of the article (at the moment only the date is used and the time of day is not taken into

[1] See http://www.darpa.mil/ipto/research/tides/.

[2] See http://www.itl.nist.gov/iaui/894.01/tests/ace/.

[3] The full set of guidelines are available as Ferro et al. (2000)

account) and the other time-point is the last time-point assigned within the same sentence. The procedure of time-stamping is as follows:

1. The text is divided into event clauses
2. All date phrases in the text are extracted
3. A date is assigned to each event clause based on either
   (a) the most recent date phrase in the same sentence, or
   (b) if this is not defined, then the date of the article.

In assigning dates various time assignment rules are used. When a date phrase is present in the sentence these rules both take into account nearby prepositions, such *on*, *after, before*, and carry out fuller specification. For example if the date phrase is simply a day of the week, then the article date is also used to derive a date-stamp that is fully specified with respect to year and position within the year. If no date phrase is present in the sentence then tense information is used to assign a time interval relative to the date of the article.

After all events have been stamped with a time, the event clauses are arranged in chronological order. The authors report scores of 77.85% correct time-stamp assignment to event clauses which have been manually (i.e. correctly) extracted from sample texts of a small trial corpus.

2.2.3. Temporal Semantic Tagging of Newswire Texts

The ultimate goal for Schilder and Habel (2001), as for ourselves, is to establish the temporal relations between all events in news articles.

In Schilder and Habel's approach temporal expressions are classified into *time-denoting expressions* that refer to a calendar or clock time and *event-denoting expressions* which refer to events. They view their goal as anchoring these temporal expression on the absolute time-line, so as to produce a linearly ordered anchored set of temporal entities; hence a time-stamp representation appears to be their target representation. For time-denoting expressions this may mean resolving indexicals (*now*, *yesterday*) or fleshing out expressions like *Thursday* to fully specified calendar dates. For event-denoting expressions a calendar time which is the time of the event must be associated with the event, possibly by extracting temporal relations which are signalled by prepositional phrases like *on Friday*. The set of temporal relations proposed is *before, after, incl, at, starts, finishes* and *excl* (equivalent to Allen (1983)'s relations).

They have developed a semantic tagging system for temporal expressions in newswire articles. The main part of their system is a Finite State Transducer (FST) based on handwritten rules. Their target language is German. The FST tags all time-denoting expressions, all verbs and an experimental version tags event-signalling nominal expressions. A semantic representation is then proposed, based on which inferences are drawn, especially about temporal relations. In its current state, the FST establishes temporal relations between times and events. The tagger was evaluated with respect to a small corpus (10 news articles) and an overall precision rate of 84.49% was achieved.

2.3. Annotating Temporal Relations

The work described in the preceding section aims at associating a calendar time with some or all events reported in a text, but none of these approaches view the identification of temporal relations as a explicit goal in its own right. These temporal relations are clearly of importance, even for time-stamping approaches. The work described in this section, as well as the approach we develop in the next section, address temporal relations directly.

2.3.1. Annotation of Intrasentential Temporal Information

Katz and Arosio (2001) aim to create a large multilingual corpus, in which intrasentential temporal relations are tagged in addition to standard morphological and syntactic features. To aid this, they have developed a language-neutral and theory-neutral method for annotating sentence internal temporal relations. With this corpus, Katz and Arosio (2001) hope to be able to automatically acquire the lexical knowledge required for determining temporal interpretation in narrative discourse.

A temporal interval is associated with each verb in the sentence; it is the temporal relations between these intervals that are of concern. The temporal interpretation should be closely linked to the syntactic context (which is of importance since it is not known beforehand to what degree the cues used by the speaker are lexical and to what degree they are grammatical). This linking is needed to keep track of both the semantic relations among times as well as the syntactic relations among the words in the sentences that refer to these times.

The authors add a layer of semantic annotation to already syntactically annotated text. The verbs in the sentence are linked via secondary edges labelled with a temporal relation. Precedence and inclusion and their duals are the possible relations. Indexical information is included by introducing the symbol $\circ$ for the speech time, which is automatically prefaced to all sentences prior to annotation.

A searchable multi-language annotated treebank has been created where each sentence is stored in a relational database with both syntactic and temporal annotations. This makes is possible to query the corpus ("Find the sentences containing a relative clause which is interpreted as temporally overlapping the main clause" (Katz and Arosio, 2001)).

This work is valuable, especially for linguists interested in the studying, cross-lingually, the complex interrelationship of lexical and syntactic mechanisms used to convey temporal relations between events in the same sentence. However, if one's goal is extraction of the full temporal content of a text, it is limited in only considering intrasentential temporal relations.

3. Annotating Temporal Information in Text

From the preceding overview of existing work on temporal information extraction it is clear that the bulk of work so far has gone into the identification of temporal referring expressions and the assignment of time-stamps to events. Only Katz and Arosio (2001) focus directly on the problem of identifying temporal relations between events, and in their case only between events in the same sentence.

In this section we start by arguing that a time-event graph, in which not all events are necessarily directly anchored on a time-line, is a superior target representation for a text to a time stamped representation. We then present the conceptual underpinning for the approach we advocate for annotating temporal information in text, followed by the details of the annotation scheme itself.

3.1. Why Annotate Temporal Relations?

Recall that a time-event graph is a graph in which the nodes are either times or events and the arcs are temporal relations. There are two principal arguments for preferring a time-graph representation to a time-stamp representation.

First, in many cases texts position events in time only by relation to other events and any attempt to coerce these events onto a time-line must either lose information, invent information, or rely on a notion of an underspecified time point constrained by temporal relations (i.e. introduce a representation of temporal relations by the back door).

Consider this example:

> After the plane crashed, a search was started. Afterwards the coast guard reported finding debris.

and assume that an earlier sentence specifies the calendrical time of the plane crash.

An approach attempting to map the information presented in this example onto a time-line is faced with the situation depicted in Figure 1.

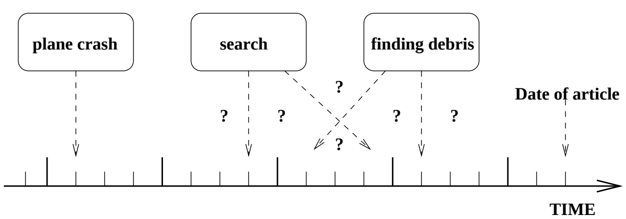

Figure 1: A Time-line Representation

While the crash event can be placed on the time-line the other two events cannot. Either time points must be guessed, or an interval be assigned. The first option is clearly not satisfactory. But if an interval is assigned the only possible interval, for both the searching and finding events is the interval from the crash till the date of the article. But if this is assigned to both events then the information about their ordering with respect to each other is lost.

A simpler representation which while not attempting to be as specific actually carries more information is shown in Figure 2.

This representation preserves the information that the searching event precedes the finding event, without forcing any early commitment to points on a time-line.

The second argument for preferring a time-event graph representation that captures event-event temporal relations as well as time-event relations is that to position events on a time-line accurately requires the extraction of event-event relational information. In the example, the placing of the

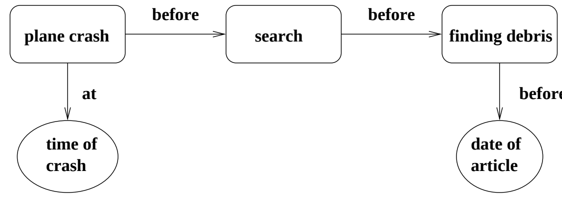

Figure 2: A Time-Event Graph Representation

searching and finding events in the interval between the plane crash and the date of the article requires the recognition that these events occurred after the crash as signalled by the words "after" and "afterwards". Without identifying the relations conveyed by these words the searching and finding events could only be positioned before the time of the article, and not following the plane crash. Thus, even if a time-stamp representation is viewed as the best target representation, achieving it requires the extraction of temporal relational information. In this case adopting a time-event graph as an intermediate representation is still a good idea, which begs the question of why it should not simply be taken as the final target representation.

3.2. Conceptualising Time

Before we describe the annotation scheme we have developed, we will very briefly explain what kind of temporal entities and relations we suppose exist. We presume the world contains the following primitives: events, states, times, temporal relations and event identity. Each primitive is described briefly below.

Events Intuitively an event is something that happens, something that one can imagine putting on a time map. Events can be ongoing or conceptually instantaneous, we do not distinguish between these. What defines an event is very much dependent on the application and domain, but generally events have to be anchorable on a time-line and they are usually conveyed in language by finite verbs or by nominalisations. Examples of events are:

> *A small single-engine plane **crashed** into the Atlantic Ocean.*
>
> *The 1996 **crash** of the TWA 747 remains unexplained.*

Times Like events, times can be viewed as having extent (intervals) or as being punctual (points). Rather than trying to reduce one perspective to the other, the focus of much of the philosophical debate on time, we shall simply treat both as *time objects*. A time object must, however, be capable of being placed on a time-line (fictional or real).

Following general convention, and the approach taken in MUC, we distinguish between two classes of time objects, DATES and TIMES, time objects which are larger or smaller than a day, respectively.

States A state is a relation between entities or the holding of an attribute of an entity which, while capable of change, is ongoing over a time span, usually longer than the time span covered by the text of interest. Examples are:

> *The plane, **which can carry four people**, ...*

The water ***is about 125 feet deep*** *in that area.*

Typically, a change of state constitutes an event. At this point we are less interested in states, and we have not taken them into account in our annotation scheme.

Temporal Relations Events stand in certain temporal relations to other events and to times. Times are temporally related to other times as well, but this phenomenon is not only very rarely explicitly expressed in text, it is also of lesser importance and is not taken into account here.

The plane crashed ***after*** *the pilot and his crew ejected.*

A small single engine plane crashed into the Atlantic Ocean ***on*** *Wednesday.*

The full set of temporal relations we suppose at present is { *included, includes, after, before, simultaneous* } . This is a minimal set, which was defined after analysing a number of newspaper articles, and can easily be expanded.

3.3. The Annotation Scheme

Given this conceptual framework, we can describe the annotation scheme we have defined. For more details see Setzer and Gaizauskas (2000).

Annotating Events Events are marked by annotating a representative in the clause conveying the event. The first choice for a representative is the head of the finite verb group. If a nominalisation conveys the event, then the head of the nominalisation serves as the representative. In the rare case of an event being conveyed by a non-finite clause, the non-finite verb is marked as the representative.

An events carries attributes for some or all of the following properties: unique event ID, event class, verb tense, verb aspect, other event to which it is related and temporal relation by which it is related, time object to which it is related and temporal relation by which it is related, the word(s) by which the temporal relation is signalled, and the ID of events it might have as an argument. For example, ignoring temporal relations for the moment:

```
A small single-engine plane
<event eid=16 class=OCCURRENCE tense=past>
    crashed
</event>
into the Atlantic Ocean about eight miles off New
Jersey
```

Annotating Times We distinguish between *simple* and *complex* time referring expressions. Simple time referring expressions refer to times directly, as in example (1). Complex time referring expressions, as in (2), refer to a point in time by relating (*after*) an interval (*17 seconds*) to an event (*hearing the sound*). The point in time referred to is the point at the end of the interval.

(1) *last Thursday*

(2) *17 seconds after hearing the sound ...*

For simple time referring expressions we annotate the whole text span conveying the time-object:

```
<timex tid=5 type=DATE calDate=12041997>
  last Thursday </timex>
```

Each time referring expression has a unique ID, an attribute flagging whether it is a time or a date, and an attribute carrying the calendar date the expression refers to.

Complex time referring expressions, like the one in example (2), include a time interval (*17 seconds*), a preposition (*after*) and an event (*hearing the sound*) or time. The way these are annotated is similar to the way events are annotated. The interval is chosen as the representative for the time referring expression and related to the event expression via the temporal relation, usually signalled by the preposition.

```
<timex tid=5 type=complex eid=3 signalID=7
relType=after> 17 seconds </timex>
<signal sid=7> after <signal>
<event eid=3> hearing</event> the sound...
```

Annotating Temporal Relations Events and times can be related to other events or times. If two events are related then one of the events carries the ID of the other as well as the temporal relation in which they stand to each other. If an event is related to a time then the event carries the ID of the time object and the temporal relation. In either case, if the relation is signalled explicitly in the text, then the ID of this signal is an attribute as well, as the following two examples illustrate.

```
All 75 people on board the Aeroflot Airbus
<event eid=4 class=OCCURRENCE tense=past
  relatedToEvent=5 eventRelType=simultaneous
  signal=7>
    died </event>
<tr_signal sid=7> when </tr_signal>
it
<event eid=5 class=OCCURRENCE tense=past >
    ploughed </event>
into a Siberian mountain.

A small single-engine plane
<event eid=9 class=OCCURRENCE tense=past
  relatedToTime=5 timeRelType=included
  signal=9>
    crashed </event>
into the Atlantic Ocean about eight miles off
New Jersey
<tr_signal sid=9> on </tr_signal>
<timex tid=5> Wednesday </timex>.
```

If the temporal relation is implicitly expressed, then the only difference is that the attribute for the signal is simply left out.

One problem with this annotation scheme is that it is not possible to relate one event to two or more other events or times, though by and large we have not found this to be a problem in annotating real text. This problem has been addressed by the TERQAS[4] workshop, which is working towards defining a general time markup language and has adopted many aspects of the current annotation scheme.

[4] See http://www.cs.brandeis.edu/ jamesp/arda/time/.

The solution proposed there is to introduce independent SGML `LINK` entities, which consume no text, to serve as relational objects tying events and times together. One event can then participate in as many links as is necessary.

4. The Pilot Study

To study the feasibility of the annotation scheme and to gain insight into the linguistic mechanisms conveying temporal information in text, we have applied the annotation scheme to a small trial corpus.

4.1. The Corpus

The trial corpus consists of 6 newswire articles taken from the New York Times, 1996, which were part of the MUC7 (MUC, 1998) training data. Basic statistics about the corpus are presented in table 1.

	sentences	words	number of annotators
text1	26	448	3
text2	18	333	2
text3	13	269	3
text4	13	213	2
text5	10	211	3
text6	13	399	3
total	93	1873	3

Table 1: The corpus

Each text was annotated by either two or three annotators, in addition to one of the authors, who produced what in the following is taken to be the ‘gold standard’ or ‘key’ annotation.

4.2. The Process of Annotation

The annotation takes place in two stages, both of which are described briefly in this section. To aid the annotator with her or his task, we have developed an annotation tool which not only allows the annotation of the information required by the scheme but which also interactively supports the annotator during the second phase, where additional temporal relations are established.

Stage I During Stage I, all event and time expressions are annotated as well as all signal expressions. Afterwards, those temporal relations that are explicitly expressed, e.g. by temporal prepositional phrases or subordinate clauses, and hold between events or events and times are established and stored as event attributes. Some implicitly expressed temporal relations are also established during this stage, for example, when events are clearly positioned in time but the signal expression has been omitted, as in *The army said Friday [...]*. In addition, *ing*-clauses without a subject can also be used to implicitly express a temporal relation between two events and are annotated during this stage.

Stage II The annotation scheme we have developed is aimed at establishing as many temporal relations in the text as possible. To relieve the burden on the annotator, and to increase the number of temporal relations annotated, we introduced stage II, which is cyclical in nature. Based on the information available, which in the beginning consists of the events, times and the temporal relations annotated in stage I, all inferences possible are drawn, according to an axiomatisation of the temporal relations *included, includes, after, before, simultaneous*. This is conducted automatically by the annotation tool which computes the deductive closure over these temporal relations. If the temporal relation between any pair of events or events and times is still unknown, the annotator is prompted for one of these [5] and, again, all possible inferences are automatically drawn. The process continues until every event-event and event-time pair in the text has been related.

4.3. The Results

In this section, we briefly describe the distribution of temporal phenomena over the trial corpus, as far as this is relevant to the issues discussed in this paper. Note that although this is a trial corpus, the results are indicative. We will not talk here about recall and precision values of the individual annotators with respect to the gold standard here – see section 5. For more information about the pilot study and its outcome see Setzer (2001).

Table 2 shows the number of event expressions, time expressions, and the number of event-event relations annotated in each text of the corpus in Stage I of the annotation process – i.e. these are the temporal relations that are explicitly expressed in the texts.

	# sentences	# words	# event expr.	# event-event relations	# event-time relations
text1	26	448	40	10	12
text2	18	333	30	10	5
text3	13	269	19	7	3
text4	13	213	27	5	0
text5	10	211	16	1	4
text6	13	399	26	13	5
total	93	1873	158	46	30

Table 2: Number of event expressions and explicit temporal relations per text

Table 3 shows for each text the number of event and time expressions in the text, the number of explicit temporal relations annotated in Stage I, the number of relations inferred from these without any further input from the annotator, the number of relations solicited from the annotator (i.e. the implicit temporal relations), and the number of inferred temporal relations overall.

4.4. Discussion

In Section 3.1. we criticised the time-stamped event sequence as a target representation on two grounds:

1. Forcing events to be placed on a time-line may result in the loss of event-event ordering information,

[5] Note that *unknown* is a possible value for a temporal relation here.

	event + time expr.	annotated ev.-ev. and ev.-time relations	inferred relations based on annotated relations	solicited rel.	total inferred relations
text1	32 + 11	10 + 12	222	124	1005
text2	26 + 5	10 + 5	122	93	380
text3	17 + 3	7 + 3	21	49	141
text4	18 + 0	5 + 0	8	45	120
text5	10 + 4	1 + 4	13	18	110
text6	24 + 5	13 + 5	107	52	514
total	127 + 28	46 + 29	493	381	2270

Table 3: Annotated, solicited, and inferred temporal relations

since the time-stamps assigned to distinct events may be identical even though we know the events occurred at separate times and know their order.

2. Event-event relational information must be extracted in order to position events on a time-line. Given this, why not choose a target representation that includes this richer information.

While both of these observations are true in general, ideally we would like to substantiate them empirically and quantitatively with respect to the trial corpus. Unfortunately we have not as yet been able to carry out the analysis for the whole corpus. However, we have chosen one text from the corpus (text6) and investigated it in detail.

To corroborate the first point above, we read text6 and, assuming perfect knowledge of the temporal information contained, then represented this information on a time-line, associating an interval with each event. In other words, without worrying about *how* the temporal information is extracted we time-stamped each event, where each time-stamp contains a start and end time expressed as calendar dates or, for at most one of the times, a symbol indicating the time is unknown.

For example, the sentences *A senior investigator looked at the wreckage Tuesday* and *Flight 800 exploded midair 20 days before Tuesday and then plunged into the ocean*[6] can be represented on a time-line as shown in Figure 3.

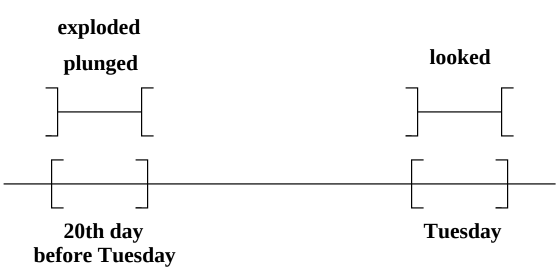

Figure 3: Example of a time-line Representation

Note that the events *exploded* and *plunged* have to be associated with the interval which encompasses the 20th day before Tuesday. We have lost the information that the plane plunged into the ocean **after** the explosion. This information can be easily represented in a time-event graph, as shown in figure 4.

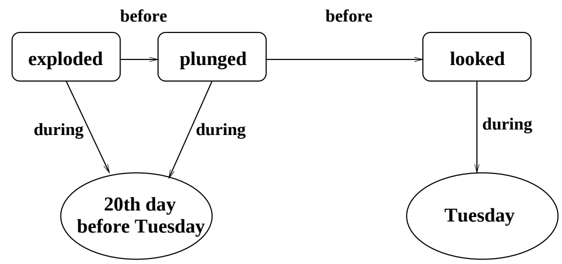

Figure 4: Example of a time-event graph Representation

Overall, 7 event-event relations that were explicitly mentioned in the text were lost in the time-line representation. While we have not performed the detailed analysis to let us say how many of the 514 inferred temporal relations in text6 are dependent on these 7 relations, it seems fair to assume that a significant number are.

To corroborate the second point we investigated how many of the 107 relations inferred for text6 from the explicitly annotated event-event and event-time relations resulted in new event-time relations involving events for which no event-time relation existed already. This corresponds to the intuitive notion of how many events are placeable on the time-line solely due to event-event relational information. For text6 we discovered that 20 of the 107 new relations were time-event relations for events for which no previous time-event relation existed. These 20 relations mentioned 4 distinct events (i.e. these 20 relations involved relating 4 events to different times, perhaps redundantly, but also potentially defining separate start and end points for intervals associated with them). Thus, 4 of the 24 events in text6 can be placed on a time-line using event-event relational information which is explicitly present in the text – positional information that otherwise would either be lost or require knowledge of implicit relations to extract.

Finally, we can make the general observation of the trial corpus that from 127 event-event relations plus 28 event-time relations, a total of 2270 additional temporal relations has been inferred. Even though we do not have the exact figure of how many of these inferred temporal relations are based on annotated event-event relations, it seems likely that the event-event relations contribute significantly to the number of relations inferred. We base this observation on the fact that there are nearly twice as many event-event relations as event-time relations annotated, and that subsequent inferences in the deductive closure calculation build on these initial relations. This observation adds weight to our claim that annotating event-event relations is important for temporal information extraction.

5. Improvements to the Annotation Process

The pilot study has shown that the interannotator agreement and the recall and precision figures need to be improved and that the burden on the annotator needs to be lessened, before the annotation scheme can be used to cre-

[6] The sentences have been slightly altered to make them more comprehensible out of context, but the temporal information they convey is the same as in the original text.

ate a larger corpus. Larger corpora will be necessary to train and evaluate temporal information extraction systems.

In Setzer and Gaizauskas (2001) we identified five main causes of low annotator precision and recall scores (with respect to the gold standard): imprecision/incompleteness of the guidelines; imperfect annotator understanding of the task; intrinsic difficulty of identifying the appropriate temporal relation in some cases; annotator fatigue; and annotator carelessness. In this section we do not address all of these problems, but focus on a number of proposals to enhance the annotation process, thereby lightening the load on annotators and increasing the accuracy of the annotations.

Pre-tagging An automatic first annotation pass could be used to reduce the amount of manual annotation and to raise recall. A part-of-speech tagger or word group parser, could be used to mark up finite verbs and signals and a time expression tagger such as Wilson et al. (2001)'s could be used to tag time referring expressions. Using the corpus as an indication, we know that a large percentage of the finite verbs will indicate events and the annotator can easily add attribute information to those or delete the mark up of mistakenly flagged verbs which do not indicate events. The high accuracy of time expression taggers mean that most of the work of tagging time expressions would be done automatically with the annotator left only to confirm details and scan for missed expressions.

Signals are a slightly different case. These are mostly prepositions and subordinating conjunctions, but a smaller number will have to be marked up. Here we have two options. We can mark up all prepositions and conjunctions and leave it to the annotator to delete inappropriate annotations, which is an easy process. Alternatively we could only automatically annotate those prepositions which are followed by a time referring expression. This approach carries the danger of not pre-annotating all signals, and the annotator, concentrating on the pre-annotated sections, might not catch all signals.

Intelligent Interaction with the Annotation Tool: Question Ordering The second phase of stage II of the annotation process, during which the deductive closure over the temporal relations is calculated and the annotator is prompted for unknown temporal relations, is problematic for the following reasons.

1. It is a long process, during which the annotator was prompted for 62 temporal relations per text on average, even for the short texts in the pilot corpus.

2. There is, for now, only marginal consistency checking and it is not possible to correct errors. Once the annotator notices that she or he made a mistake earlier in the process, then the whole stage II annotation process has to be restarted.

One possible solution for the first problem would be to optimise the order in which unknown temporal relations are prompted for. As we explained in section 4.2., after each temporal relation solicited from the user, all possible inferences are drawn. The larger the number of the inferences, the smaller the number of remaining unknown temporal relations will be. The following simple example illustrates the effect non-optimal soliciting can have. Imagine four events, forming a 'precedence chain':

$$e_1 < e_2 < e_3 < e_4$$

Imagine also that the link between e_2 and e_3 is missing in the response:

$$e_1 < e_2 \qquad e_3 < e_4$$

If the first question establishes the temporal relation holding between e_2 and e_3, then all other temporal relations can be inferred, based on the transitivity of before. The temporal model can be completed with one question. However, the order of questioning could be very different, establishing the temporal relations between e_1 and e_4, then between e_1 and e_3, e_2 and e_4 and then between e_2 and e_3. In this case four questions are asked to establish the relations holding between them.

Thus, question order can be important in determining how many questions the annotator ultimately gets asked. Clearly, one wants to minimise the number of questions asked, but it is not clear (to us) whether there is a question order that is guaranteed to minimise this number, and if so how to determine it. We propose to investigate initially a naive approach in which given two temporally-ordered event chains we first ask questions which attempt to link their end points, simply on the grounds that such questions could lead to maximal gains. However, considerably more empirical and theoretical investigation needs to be carried out here.

Intelligent Interaction with the Annotation Tool: Correcting Mistakes The second point requires a more elaborate solution. Once an incorrect temporal relation has been added an indeterminate number of further incorrect inferences may have been drawn on the basis of it. Two solutions suggest themselves:

1. Provide the possibility of check-pointing, i.e. saving intermediate stages to which the annotator can return when an error has been detected. This could be done automatically after each new user-solicited relation is added. This has the advantage of being easy to implement but the disadvantage of erasing possibly correct temporal relations added after the error, but independently of it, with the consequence that work that will have to be redone unnecessarily.

2. Implement a sort of truth maintenance system (Doyle, 1987; de Kleer, 1987), whereby only the incorrect temporal relation and those temporal relations which were inferred from it are deleted. This has the advantage of minimising the amount of work the annotator needs to redo unnecessarily, but the disadvantage of being more complex to implement.

Clearly the second solution is the better in the long run, as annotator effort is the chief quantity to conserve. We are working on solution whereby all temporal relations added to the temporal fact database record with them a justification which includes a reference to any facts from which they have been derived. Removing a temporal fact f then becomes a recursive procedure which begins with a search for all facts f' whose justification mentions f followed by a recursive call to delete f'. This will ensure that all dependents of f will be removed, while not touching any facts, solicited or derived, that may have been added after f in the annotation process, but which are logically independent of it.

6. Conclusion

We have argued that when extracting temporal information from texts a target representation, such as a time-event graph, which explicitly admits event-event temporal relations as well as time-event relations, is superior to one which does not, such as a time-stamped event sequence representation. In essence the arguments are that a time-stamp representation forces overspecification leading to information loss, and that event-event relations must be extracted even if a time-stamp representation is the target, and hence might as well be retained.

We also described the annotation scheme we have developed, which enables us to annotate temporal relations as well as events and time referring expressions, thus providing the necessary information to build time-event graphs for texts. A trial corpus which we constructed based on this scheme was described and used to corroborate the argument in support of the time-event graph approach.

One potential practical argument against the time-event graph approach is that building annotated resources capturing the required information is costly and error-prone. In the final section of the paper we introduced ideas for improving quality and reducing effort in the annotation process, improvements which we hope will make future larger scale application of the annotation scheme feasible.

7. References

J.F. Allen. 1983. Maintaining Knowledge About Temporal Intervals. *Communications of the ACM*, 26:832–843.

J. de Kleer. 1987. An Assumption-based TMS. In M.L. Ginsberg, editor, *Readings in Nonmonotonic Reasoning*, pages 280–298. Morgan Kaufman Publishers, Los Altos, California.

J. Doyle. 1987. A Truth Maintenance System. In M.L. Ginsberg, editor, *Readings in Nonmonotonis Reasoning*, pages 259–279. Morgan Kaufman Publishers, Los Altos, California.

L. Ferro, I. Mani, B. Sundheim, and G. Wilson. 2000. TIDES Temporal Annotation Guidelines. Technical Report MTR 00W0000094, The MITRE Corporation, October. Draft-Version 1.0.

E. Filatova and E. Hovy. 2001. Assigning Time-Stamps to Event-Clauses. In *Proceedings of ACL-EACL 2001, Workshop for Temporal and Spatial Information Processing*, pages 88–95, Toulouse.

G. Katz and F. Arosio. 2001. The Annotation of Temporal Information in Natural Language Sentences. In *Proceedings of ACL-EACL 2001, Workshop for Temporal and Spatial Information Processing*, pages 104–111, Toulouse. Association for Computational Linguistics.

1998. *Proceedings of the Seventh Message Understanding Conference (MUC-7)*. Morgan Kaufman. http://www.itl.nist.gov/iaui/894.02/related_projects/muc/.

F. Schilder and C. Habel. 2001. From Temporal Expressions to Temporal Information: Semantic Tagging of News Messages. In *Proceedings of ACL-EACL 2001, Workshop for Temporal and Spatial Information Processing*, pages 65–72, Toulouse.

A. Setzer and R. Gaizauskas. 2000. Annotating Events and Temporal Information in Newswire Texts. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC2000)*, pages 1287–1293, Athens, Greece.

A. Setzer and R. Gaizauskas. 2001. A Pilot Study On Annotating Temporal Relations In Text. In *Proceedings of ACL-EACL 2001, Workshop for Temporal and Spatial Information Processing*, pages 73–80, Toulouse, France.

A. Setzer. 2001. *Temporal Information in Newswire Articles: An Annotation Scheme and Corpus Study*. Ph.D. thesis, University of Sheffield.

G. Wilson and I. Mani. 2000. Robust Temporal Processing of News. In *Proceedings of the 38th Meeting of the Association of Computational Linguistics (ACL 2000)*, pages 69–76, Hong Kong, October.

G. Wilson, I. Mani, B. Sundheim, and L. Ferro. 2001. A Multilingual Approach to Annotating and Extracting Temporal Information. In *Proceedings of ACL-EACL 2001, Workshop for Temporal and Spatial Information Processing*, pages 81–87, Toulouse.