

# An Intermediate Representation for the Interpretation of Temporal Expressions

Pawel Mazur and Robert Dale

Centre for Language Technology

Macquarie University

NSW 2109 Sydney Australia

{mpawel,rdale}@ics.mq.edu.au

## Abstract

The interpretation of temporal expressions in text is an important constituent task for many practical natural language processing tasks, including question-answering, information extraction and text summarisation. Although temporal expressions have long been studied in the research literature, it is only more recently, with the impetus provided by exercises like the ACE Program, that attention has been directed to broad-coverage, implemented systems. In this paper, we describe our approach to intermediate semantic representations in the interpretation of temporal expressions.

## 1 Introduction

In this paper, we are concerned with the interpretation of temporal expressions in text: that is, given an occurrence in a text of an expression like that marked in italics in the following example, we want to determine what point in time is referred to by that expression.

- (1) We agreed that we would meet at *3pm on the first Tuesday in November*.

In this particular case, we need to make use of the context of utterance to determine which November is being referred to; this might be derived on the basis of the date stamp of the document containing this sentence. Then we need to compute the full time and date the expression corresponds to. If the utterance in (1) was produced, say, in July 2006, then we might expect the interpretation to be equivalent to the ISO-format expression 2006-11-

07T15:00.<sup>1</sup> The derivation of such interpretation was the focus of the TERN evaluations held under the ACE program. Several teams have developed systems which attempt to interpret both simple and much more complex temporal expressions; however, there is very little literature that describes in any detail the approaches taken. This may be due to a perception that such expressions are relatively easy to identify and interpret using simple patterns, but a detailed analysis of the range of temporal expressions that are covered by the TIDES annotation guidelines demonstrates that this is not the case. In fact, the proper treatment of some temporal expressions requires semantic and pragmatic processing that is considerably beyond the state of the art.

Our view is that it is important to keep in mind a clear distinction between, on the one hand, the conceptual model of temporal entities that a particular approach adopts; and, on the other hand, the specific implementation of that model that might be developed for a particular purpose. In this paper, we describe both our underlying framework, and an implementation of that framework. We believe the framework provides a basis for further development, being independent of any particular implementation, and able to underpin many different implementations. By clearly separating the underlying model and its implementation, this also opens the door to clearer comparisons between different approaches.

We begin by summarising existing work in the area in Section 2; then, in Section 3, we describe our underlying model; in Section 4, we describe how this model is implemented in the DANTE

---

<sup>1</sup>Clearly, other aspects of the document context might suggest a different year is intended; and we might also add the time zone to this value.

system.<sup>2</sup>

## 2 Relation to Existing Work

The most detailed system description in the published literature is that of the Chronos system from ITC-IRST (Negri and Marseglia, 2005). This system uses a large set of hand-crafted rules, and separates the recognition of temporal expressions from their interpretation. The ATEL system developed by the Center for Spoken Language Research (CSLR) at University of Colorado (see (Hacioglu et al., 2005)) uses SVM classifiers to detect temporal expressions. Alias-i's LingPipe also reported results for extraction, but not interpretation, of temporal expressions at TERN 2004.

In contrast to this collection of work, which comes at the problem from a now-traditional information extraction perspective, there is also of course an extensive prior literature on the semantic of temporal expressions. Some more recent work attempts to bridge the gap between these two related enterprises; see, for example, Hobbs and Pan (2004).

## 3 The Underlying Model

We describe briefly here our underlying conceptual model; a more detailed description is provided in (Dale and Mazur, 2006).

### 3.1 Processes

We take the ultimate goal of the interpretation of temporal expressions to be that of computing, for each temporal expression in a text, the point in time or duration that is referred to by that expression. We distinguish two stages of processing:

**Recognition:** the process of identifying a temporal expression in text, and determining its extent.

**Interpretation:** given a recognised temporal expression, the process of computing the value of the point in time or duration referred to by that expression.

In practice, the processes involved in determining the extent of a temporal expression are likely to make use of lexical and phrasal knowledge that mean that some of the semantics of the expression can already be computed. For example, in

<sup>2</sup>DANTE stands for Detection and Normalisation of Temporal Expressions.

order to identify that an expression refers to a day of the week, we will in many circumstances need to recognize whether one of the specific expressions  $\{\textit{Monday}, \textit{Tuesday}, \dots, \textit{Sunday}\}$  has been used; but once we have recognised that a specific form has been used, we have effectively computed the semantics of that part of the expression.

To maintain a strong separation between recognition and interpretation, one could simply recompute this partial information in the interpretation phase; this would, of course, involve redundancy. However, we take the view that the computation of partial semantics in the first step should not be seen as violating the strong separation; rather, we distinguish the two steps of the process in terms of the extent to which they make use of contextual information in computing values. Then, recognition is that phase which makes use only of expression-internal information and preposition which precedes the expression in question; and interpretation is that phase which makes use of arbitrarily more complex knowledge sources and wider document context. In this way, we motivate an intermediate form of representation that represents a 'context-free' semantics of the expression.

The role of the recognition process is then to compute as much of the semantic content of a temporal expression as can be determined on the basis of the expression itself, producing an intermediate partial representation of the semantics. The role of the interpretation process is to 'fill in' any gaps in this representation by making use of information derived from the context.

### 3.2 Data Types

We view the temporal world as consisting of two basic types of entities, these being **points in time** and **durations**; each of these has an internal hierarchical structure. It is convenient to represent these as feature structures like the following:<sup>3</sup>

$$(2) \quad \left[ \begin{array}{l} \textit{point} \\ \\ \textit{DATE} \left[ \begin{array}{ll} \textit{DAY} & 11 \\ \textit{MONTH} & 6 \\ \textit{YEAR} & 2005 \end{array} \right] \\ \\ \textit{TIME} \left[ \begin{array}{ll} \textit{HOUR} & 3 \\ \textit{MINUTE} & 00 \\ \textit{AMPM} & pm \end{array} \right] \end{array} \right]$$

<sup>3</sup>For reasons of limitations of space, we will ignore durations in the present discussion; their representation is similar in spirit to the examples provided here.

Our choice of attribute–value matrices is not accidental; in particular, some of the operations we want to carry out on the interpretations of both partial and complete temporal expressions can be conveniently expressed via unification, and this representation is a very natural one for such operations.

This same representation can be used to indicate the interpretation of a temporal expression at various stages of processing, as outlined below. In particular, note that temporal expressions differ in their **explicitness**, i.e. the extent to which the interpretation of the expression is explicitly encoded in the temporal expression; they also differ in their **granularity**, i.e. the smallest temporal unit used in defining that point in time or duration. So, for example, in a temporal reference like *November 11th*, interpretation requires us to make explicit some information that is not present (that is, the year); but it does not require us to provide a time, since this is not required for the granularity of the expression.

In our attribute–value matrix representation, we use a special NULL value to indicate granularities that are not required in providing a full interpretation; information that is not explicitly provided, on the other hand, is simply absent from the representation, but may be added to the structure during later stages of interpretation. So, in the case of an expression like *November 11th*, the recognition process may construct a partial interpretation of the following form:

$$(3) \quad \left[ \begin{array}{l} \textit{point} \\ \text{DATE} \left[ \begin{array}{ll} \text{DAY} & 11 \\ \text{MONTH} & 6 \end{array} \right] \\ \text{TIME} \quad \text{NULL} \end{array} \right]$$

The interpretation process may then monotonically augment this structure with information from the context that allows the interpretation to be made fully explicit:

$$(4) \quad \left[ \begin{array}{l} \textit{point} \\ \text{DATE} \left[ \begin{array}{ll} \text{DAY} & 11 \\ \text{MONTH} & 6 \\ \text{YEAR} & 2006 \end{array} \right] \\ \text{TIME} \quad \text{NULL} \end{array} \right]$$

The representation thus very easily accommodates relative underspecification, and the potential for further specification by means of unification, although our implementation also makes use of other operations applied to these structures.

## 4 Implementation

### 4.1 Data Structures

We could implement the model above directly in terms of recursive attribute–value structures; however, for our present purposes, it turns out to be simpler to implement these structures using a string-based notation that is deliberately consistent with the representations for values used in the TIMEX2 standard (Ferro et al., 2005). In that notation, a time and date value is expressed using the ISO standard; uppercase Xs are used to indicate parts of the expression for which interpretation is not available, and elements that should not receive a value are left null (in the same sense as our NULL value above). So, for example, in a context where we have no way of ascertaining the century being referred to, the TIMEX2 representation of the value of the underlined temporal expression in the sentence *We all had a great time in the '60s* is simply VAL="XX6".

We augment this representation in a number of ways to allow us to represent intermediate values generated during the recognition process; these extensions to the representation then serve as means of indicating to the interpretation process what operations need to be carried out.

#### 4.1.1 Representing Partial Specification

We use lowercase xs to indicate values that the interpretation process is required to seek a value for; and by analogy, we use a lowercase t rather than an uppercase T as the date–time delimiter in the structure to indicate when the recogniser is not able to determine whether the time is am or pm. This is demonstrated in the following examples; T-VAL is the attribute we use for intermediate TIMEX values produced by the recognition process.

- (5) a. We'll see you in *November*.  
b. T-VAL="xxxx-11"
- (6) a. I expect to see you at *half past eight*.  
b. T-VAL="xxxx-xx-xxt08:59"
- (7) a. I saw him back in '69.  
b. T-VAL="xx69"
- (8) a. I saw him back in the '60s.  
b. TVAL="xx6"

#### 4.1.2 Representing Relative Specification

To handle the partial interpretation of relative date and time expressions at the recognition stage, we

use two extensions to the notation. The first provides for simple arithmetic over interpretations, when combined with a reference date determined from the context:

- (9) a. We'll see you *tomorrow*.  
b. T-VAL=" +0000-00-01 "
- (10) a. We saw him *last year*.  
b. T-VAL=" -0001 "

The second provides for expressions where a more complex computation is required in order to determine the specific date or time in question:

- (11) a. We'll see him *next Thursday*.  
b. T-VAL=" >D4 "
- (12) a. We saw him *last November*.  
b. T-VAL=" <M11 "

## 4.2 Processes

For the recognition process, we use a large collection of rules written in the JAPE pattern-matching language provided within GATE (see (Cunningham et al., 2002)). These return intermediate values of the forms described in the previous section. Obviously other approaches to recognizing temporal expressions and producing their intermediate values could be used; in DANTE, there is also a subsequent check carried out by a dependency parser to ensure that we have captured the full extent of the temporal expression.

DANTE's interpretation process then does the following. First it determines if the candidate temporal expression identified by the recogniser is indeed a temporal expression; this is to deal with cases where a particular word or phrase annotated by the recognizer (such as *time*) can have both temporal or non-temporal interpretations. Then, for each candidate that really is a temporal expression, it computes the interpretation of that temporal expression.

This second step involves different operations depending on the type of the intermediate value:

- Underspecified values like `xxxx-11` are combined with the reference date derived from the document context, with temporal directionality (i.e., is this date in the future or in the past?) being determined using tense information from the host clause.
- Relative values like `+0001` are combined with the reference date in the obvious manner.

- Relative values like `>D4` and `<M11` make use of special purpose routines that know about arithmetic for days and months, so that the correct behaviour is observed.

## 5 Conclusions

We have sketched an underlying conceptual model for temporal expression interpretation, and presented an intermediate semantic representation that is consistent with the TIMEX2 standard. We are making available a corpus of examples tagged with these intermediate representations; this corpus is derived from the nearly 250 examples in the TIMEX2 specification, thus demonstrating the wide coverage of the representation. Our hope is that this will encourage collaborative development of tools based on this framework, and further development of the conceptual framework itself.

## 6 Acknowledgements

We acknowledge the support of DSTO, the Australian Defence Science and Technology Organisation, in carrying out the work described here.

## References

- H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. 2002. GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of the 40th Anniversary Meeting of the ACL*.
- R. Dale and P. Mazur. 2006. Local semantics in the interpretation of temporal expressions. In *Proceedings of the Coling/ACL2006 Workshop on Annotating and Reasoning about Time and Events*.
- L. Ferro, L. Gerber, I. Mani, B. Sundheim, and G. Wilson. 2005. TIDES 2005 Standard for the Annotation of Temporal Expressions. Technical report, MITRE, September.
- K. Hacioglu, Y. Chen, and B. Douglas. 2005. Automatic Time Expression Labeling for English and Chinese Text. In Alexander F. Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing, 6th International Conference, CICLing'05*, LNCS, pages 548–559. Springer.
- Jerry R. Hobbs and Feng Pan. 2004. An ontology of time for the semantic web. *ACM Transactions on Asian Language Information Processing*, 3(1), March.
- M. Negri and L. Marseglia. 2005. Recognition and Normalization of Time Expressions: ITC-irst at TERN 2004. Technical Report WP3.7, Information Society Technologies, February.