

Local Semantics in the Interpretation of Temporal Expressions

Robert Dale* and Paweł Mazur†

Centre for Language Technology
Macquarie University, NSW 2109, Sydney, Australia

*Robert.Dale@mq.edu.au,

†mpawel@ics.mq.edu.au

Abstract

Work on the interpretation of temporal expressions in text has generally been pursued in one of two paradigms: the formal semantics approach, where an attempt is made to provide a well-grounded theoretical basis for the interpretation of these expressions, and the more pragmatically-focused approach represented by the development of the TIMEX2 standard, with its origins in work in information extraction. The former emphasises formal elegance and consistency; the latter emphasises broad coverage for practical applications. In this paper, we report on the development of a framework that attempts to integrate insights from both perspectives, with the aim of achieving broad coverage of the domain in a well-grounded manner from a formal perspective. We focus in particular on the development of a compact notation for representing the semantics of underspecified temporal expressions that can be used to provide component-level evaluation of systems that interpret such expressions.

1 Introduction

Obtaining a precise semantic representation for utterances related to time is interesting both from a theoretical point of view, as there are many complex phenomena to be addressed, and for purely practical applications such as information extraction, question answering, or the ordering of events on a timeline.

In the literature, work on the interpretation of temporal expressions comes from two directions.

On the one hand, work in formal semantics (see, for example, (Pratt and Francez, 2001)) aims to provide a formally well-grounded approach to the representation of the semantics of these expressions, but such approaches are difficult to scale up to the broad coverage required for practical applications; on the other hand, work that has its roots in information extraction, while it emphasises broad coverage, often results in the use of ad hoc representations. The most developed work in this direction is focused around the TimeML markup language¹ (described, for example, in (Pustejovsky et al., 2003) and in the collection edited by Mani et al. (2005)).

Some work attempts to bring these two traditions together: notable in this respect is Schilder's (2004) work on temporal expressions in German newswire text, and Hobbs and Pan's (2004) work on the axiomatisation in terms of OWL-Time. Sagueete et al. (2002) present an approach that views time expressions as anaphoric references.

We take the view that an important step towards a truly broad coverage yet semantically well-founded approach is to recognize that there is a principled distinction to be made between the interpretation of the semantics of a temporal expression devoid of its context of use, and the fuller interpretation of that expression when the context is taken into account. The first of these, which we refer to here as the **local semantics** of a temporal expression, should be derivable in a compositional manner from the components of the expression; determining the value of the second, which we refer to as the **global semantics** of the expression, may require arbitrary inference and reason-

¹Note that with TimeML one can annotate not only temporal expressions, but also events and relations between events and temporal expressions.

ing. Such a distinction is implicit in other accounts: Schilder’s (2004) use of lambda expressions allows representation of partially specified temporal entities, and the temporary variables that Negri and Marseglia (2005) construct during the interpretation of a given temporal expression capture something of the same notion.

Our proposal here is to reify this level of intermediate representation based on a formalization in terms of recursive attribute–value matrices. This has two distinct advantages: it provides a convenient representation of underspecification, and it leads naturally to a compositional approach to the construction of the semantics of temporal expressions via unification. We also provide a compact encoding of this representation that is essentially an extension of the existing TIMEX2 representation for temporal expressions. This brings the advantages that (a) existing tools and machinery for evaluation can be used to determine how well a given implementation derives these local semantic values; and (b) performance in the determination of local semantics and global semantics can be tested independently. To ensure breadth of coverage, we have developed our representation on the basis of the 256 examples of temporal expressions provided in the TIMEX2 guidelines (Ferro et al., 2005). To make it possible to compare systems on their performance in producing these intermediate representations, we make available this set of examples annotated in-line with the representations described here.

The rest of this paper is structured as follows. In Section 2, we describe the architecture of DANTE, a system which embodies our approach to the detection and normalisation of temporal expressions; in particular, we focus on the architecture employed in this approach, and on the particular levels of representation that it makes use of. In Section 3, we argue for an intermediate representational level that captures the semantics of temporal expressions independent of the context of their interpretation, and introduce the idea of using recursive attribute–value matrices to represent the semantics of temporal expressions. In Section 4, we provide an encoding of these attribute–value matrices in a compact string-based representation that is effectively an extension of the ISO-based date–time format representations used in the TIMEX2 standard, thus enabling easy evaluation of system performance using existing tools. In Section 5

we discuss how the approach handles constructions that contain one TIMEX embedded within another. Finally, in Section 6 we draw some conclusions and point to future work.

2 The DANTE System

2.1 Processing Steps

In our work, our goal is very close to that for which the TIMEX2 standard was developed: we want to annotate each temporal expression in a document with an indication of its interpretation, in the form of an extended ISO-format date and time string, normalised to some time zone. So, for example, suppose we have the following italicised temporal expression in an email message that was sent from Sydney on Monday 10th April 2006:

- (1) I expect that we will be able to present this at the meeting on *Friday at 11am*.

In the context of our application, this temporal expression should be marked up as follows:

- (2) <TIMEX2 VAL="2006-04-14T01:00GMT">
Friday at 11am</TIMEX2>

We have to do three things to achieve the desired result:

- First, we have to detect the extent of the temporal expression in the text. We refer to this process as **temporal expression recognition**.
- Then, we have to use information from the document context to turn the recognized expression into a fully specified date and time. We refer to this as **temporal expression interpretation**.
- Finally, we have to normalise this fully specified date and time to a predefined time zone, which in the case of the present example is Greenwich Mean Time. We refer to this as **temporal expression normalisation**.²

²Note that this third step is not required by the TIMEX guidelines, but is an additional requirement in the context of our particular application. This also means that our use of the term ‘normalisation’ here is not consistent with the standard usage in the TIMEX context; however, we would argue that our distinction between interpretation and normalisation describes more accurately the nature of the processes involved here.

We observe that, at the time that the extent of a temporal expression within a text is determined, it is also possible to derive some semantic representation of that expression irrespective of the wider context within which it needs to be interpreted: for example, by virtue of having recognized an occurrence of the string *Friday* in a text, we already know that this is a reference to a specific day of the week. Most existing systems for the interpretation of temporal expressions probably make use of such a level of representation. Schilder’s (2004) approach captures the semantics here in terms of a lambda expression like $\lambda x \text{Friday}(x)$; Negri and Marseglia (2005) capture information at this stage of processing via a collection of temporary attributes.

In our system, each of the three steps above corresponds to a distinct processing component in the DANTE system architecture. These components communicate in terms of a number of distinct representations, which we now describe.

2.2 The Text

This level of representation corresponds simply to the strings that constitute temporal expressions in text. These are understood to be linguistic constructions whose referents are entities in the temporal domain: either points in time, or periods of time. In the above example, the text representation is simply the string *Friday at 11am*.

2.3 Local Semantics

We use this term to refer to a level of representation that corresponds to the semantic content that is derivable directly from the text representation; in the case of temporal expressions that are arguments to prepositions, this includes the interpretation of the preposition. Such representations are often incomplete, in that they do not denote a particular point or period on the time line; however, usually they do partially specify points or periods, and constrain the further interpretation of the string.

2.4 In-Document Semantics

We use this term to refer to the fully explicit interpretation of the text string, to the extent that this can be determined from the document itself, in conjunction with any metadata associated with the document. This level of representation corresponds to the information encoded in the attributes

of the TIMEX2 tag as defined in the TIMEX guidelines.

2.5 Global Semantics

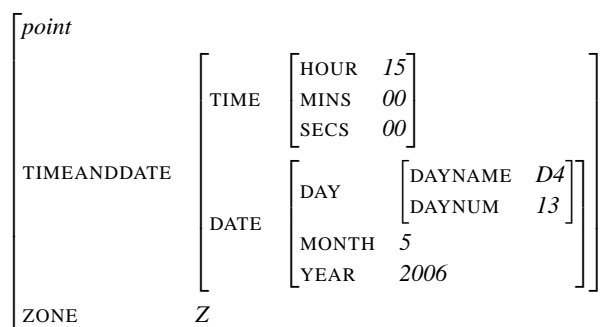
The TIMEX guidelines do not have anything to say beyond the representation described in the previous section. In our application, however, we are also required to normalise all temporal expressions to a specific time zone. This requires that some further temporal arithmetic be applied to the semantics of the found expressions. To calculate this, we simply have to determine the difference between the time zone of the document containing the temporal reference and the target time zone, here Greenwich Mean Time. The document may not always be explicitly marked with information about the time zone of its creation; in such cases, this has to be inferred from information about the location of the author or sender of the message.

3 Representing Temporal Expressions

In this section, we describe a conceptualisation of the semantics of temporal expressions in terms of recursive attribute–value matrices.

3.1 Temporal Entities

As is conventional in this area of research, 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. We can represent these in the following manner:³



The example above corresponds to the semantics of the temporal expression *3pm Thursday 13th May 2006 GMT*; in the ISO date and time format used in the TIMEX2 standard, this would be written as follows:

$$(3) \quad 2006-05-13T15:00:00Z$$

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

Each atomic feature in the attribute–value structure thus corresponds to a specific position in the ISO format date–time string.

3.2 Underspecification

Of course, very few temporal expressions in text are fully specified. The attribute–value matrix representation makes it easy to represent the content of underspecified temporal expressions. For example, the content of the temporal expression *Thursday* in a sentence like *We will meet on Thursday* can be expressed as follows:

$$\left[\begin{array}{c} \textit{point} \\ \text{TIMEANDDATE} \left[\text{DATE} \left[\text{DAY} \left[\text{DAYNAME} \textit{D4} \right] \right] \right] \right] \end{array} \right]$$

On the other hand, a reference to *13th May* in a sentence like *We will meet on 13th May* has this representation:

$$\left[\begin{array}{c} \textit{point} \\ \text{TIMEANDDATE} \left[\text{DATE} \left[\begin{array}{c} \text{DAY} \left[\text{DAYNUM} \textit{13} \right] \\ \text{MONTH} \textit{05} \end{array} \right] \right] \right] \end{array} \right]$$

In the cases just described, the semantic representation corresponds to the entire temporal noun phrase in each case. The same form of representation is easy to use in a compositional semantic framework: each constituent in a larger temporal expression provides a structure that can be unified with the structures corresponding to the other constituents of the expression to provide a semantics for the expression as a whole. The values of the atomic elements in such an expression come from the lexicon; multiword sequences that are best considered atomic (such as, for example, idioms) can also be assigned semantic representations in the same way. The value of a composite structure is produced by unifying the values of its constituents. Unifying the two structures above, for example, gives us the following representation for *Thursday 13th May*:⁴

$$\left[\begin{array}{c} \textit{point} \\ \text{TIMEANDDATE} \left[\text{DATE} \left[\begin{array}{c} \text{DAY} \left[\begin{array}{c} \text{DAYNAME} \textit{D4} \\ \text{DAYNUM} \textit{13} \end{array} \right] \\ \text{MONTH} \textit{05} \end{array} \right] \right] \right] \end{array} \right]$$

So, these structures provide a convenient representation for what we have referred to above as the

⁴For simplicity here we assume that the syntactic structure of such an expression is captured by the context-free grammar rule 'NP → NP NP'. Other treatments are possible.

local semantics of a temporal expression, and correspond to the output of the recognition stage of our processing architecture.

3.3 Interpretation

We can now define the task of interpretation in terms of the content of these structures. We assume a **granularity ordering** over what we might think of as the **defining attributes** in a temporal representation:

$$(4) \quad \text{year} > \text{month} > \text{daynum} > \text{hour} > \text{minute} > \text{second}$$

These are, of course, precisely the elements that are represented explicitly in an ISO date–time expression.

Interpretation of a partially specified temporal expression then requires ensuring that there is a value for every defining attribute that is of greater granularity than the smallest granularity present in the partially specified representation. We refer to this as the **granularity rule** in interpretation.

In the case of the example in the previous section, the granularity rule tells us that in order to compute the full semantic value of the expression we have to determine a value for YEAR, but not for HOUR, MINS or SECS. This interpretation process may require a variety of forms of reasoning and inference, as discussed below, and is qualitatively different from the computation of the local semantics.

In the context of our application, a third stage, the normalisation process, then requires taking the further step of adding a ZONE attribute with a specific value, and translating the rest of the construction into this time zone if it represents a time in another time zone.

4 A Compact Encoding

The structures described in the previous section are relatively unwieldy in comparison to the simple string structures used as values in the TIMEX standard. To enable easy evaluation of a system's ability to construct these intermediate semantic representations, we would like to use a representation that is immediately usable by existing evaluation tools. To achieve this goal, we define a number of extensions to the standard TIMEX2 string representation for values of the VAL attribute; these extensions allow us to capture the range of distinctions we need. To save space, we

also use these representations here to show the coverage of the annotation scheme that results.

In our implementation, we represent the local semantic content via an additional set of attributes on TIMEX elements that mirrors exactly the set of attributes used by the TIMEX2 standard: thus we have T-VAL, T-ANCHOR_VAL and so on. This means that markup applied to a text distinguishes intermediate and final semantic values, making it possible to evaluate on just intermediate values, just final values, or both. In what follows, we will also use these intermediate attributes to make clear which level of representation is under discussion.

4.1 Partially Specified Dates and Times

As noted above, many references to dates or times are not fully specified in a text, with the result that some parts will have to be computed from the context during the interpretation stage. Typical examples are as follows:

- (5) a. We'll see you in *November*.
 b. I expect to see you at *half past eight*.

In the recursive attribute–value notation introduced above, the missing information in each case corresponds to those features that are absent in the structure as determined by the granularity rule introduced in Section 3.3.

In our string-based notation, we use lowercase *x*s to indicate those elements for which a value needs to be found, but which are not available at the time the local semantics are computed; and we capture the granularity requirement by omitting from the string representation those elements that do not require a value.⁵ Table 1 provides a range of examples that demonstrate various forms of underspecification.

A lowercase *x* thus corresponds to a variable. By analogy with this extension, we also use a lowercase *t* instead of the normal ISO date–time separator of *T* to indicate that the time may need further specification: consider the third and fourth examples in Table 1, where it is not clear whether the time specified is a.m. or p.m.

For partially-specified dates and times, the string-based encoding thus both captures the local

⁵Note that this does not mean the same thing as the use of an uppercase *X* in the TIMEX2 guidelines: an uppercase *X* means effectively that no value can be determined. Of course, if no value can be found for a variable element during the interpretation process, then the corresponding lowercase *x* will be replaced by an uppercase *X*.

String	Representation
9 pm	xxxx-xx-xxT21
11:59 pm	xxxx-xx-xxT23:59
eleven in the morning	xxxx-xx-xxT11:00
ten minutes to 3	xxxx-xx-xxt02:50
15 minutes after the hour	xxxx-xx-xxtxx:15
the nineteenth	xxxx-xx-19
January 3	xxxx-01-03
November	xxxx-11
summer	xxxx-SU
'63	xx63
the '60s	xx6

Table 1: Underspecified Dates and Times

semantic content of the temporal expression, and provides a specification of what information the interpretation process has to add. If the temporal focus is encoded in the same form of representation, then producing the final interpretation is often a simple process of merging the two structures, with the values already specified in the intermediate representation taking precedence over those in the representation of the temporal focus. Expressions involving references to named months require a decision as to whether to look for the next or previous instance of the month, typically determined by the tense of the major clause containing the reference.

4.2 Representing Weekdays

In recognition that the year-based calendar and the week-based calendar are not aligned, our intermediate representation embodies a special case borrowed from the TIMEX2 notation for days of the week that require context for their specification. Consider example (6a), uttered on Friday 14th April 2006; the intermediate semantic representation is provided in example (6b), and the final interpretation is provided in example (6c).

- (6) a. We left on *Tuesday*.
 b. T-VAL="D2"
 c. VAL="2006-04-11"

This is not as convenient as the ISO-like encoding, and requires special case handling in the interpreter; however, a more comprehensive single representation would require abandoning the ISO-like encoding and the benefits it brings, so we choose to use the two formats in concert.

String	Representation
today	+0000-00-00
tomorrow	+0000-00-01
yesterday	-0000-00-01
five days ago	-0000-00-05
last month	-0000-01
last summer	-0001-SU
two weeks ago	-0000-W02
this weekend	+0000-W00-WE
this year	+0000
three years ago	-0003
the next century	+01

Table 2: Relative dates in ISO-like format.

The same notation supports references to parts of specific days, as presented in example (7).

- (7)
- a. We left on *Tuesday morning*.
 - b. T-VAL= "D2TMO "
 - c. VAL= "2006-04-11TMO "

4.3 Relative Dates and Times

A relative date or time reference is one that requires a calendar arithmetic operation to be carried out with respect to some temporal focus in the text. Typical examples are as follows:

- (8)
- a. We'll see him *tomorrow*.
 - b. We saw him *last year*.
 - c. We'll see him *next Thursday*.
 - d. We saw him *last November*.

We distinguish three subtypes here: relative dates and times whose local semantics can be expressed in an ISO-like format; relative references to days and months by name; and less specific references to past, present and future times.

For the first of these, we extend the ISO format with a preceding '+' or '-' to indicate the direction from the current temporal focus. Some examples of dates are provided in Table 2, and some examples of date-time combinations are provided in Table 3. Note the both the date and time elements in a relative reference can be independently either absolute or relative: compare the representations for *in six hours time* and *at 6am today*.

This representation leads to a very intuitive coordinate-based arithmetic for computing the final semantic interpretation of a given expression: the interpreter merely adds the temporal focus and

String	Representation
sixty seconds later	+0000-00-00T+00:00:60
five minutes ago	+0000-00-00T-00:05
in six hours time	+0000-00-00T+06:00
at 6 a.m. today	+0000-00-00T06:00
last night	-0000-00-01TNI

Table 3: Relative times in ISO-like format.

String	Representation
last Monday	<D1
next Wednesday	>D3
last March	<M03
next March	>M03

Table 4: Relative References to Days and Months

the intermediate value element-by-element from the smallest unit upwards, using carry arithmetic where appropriate.

Relative references to named days and months require a different treatment, in line with the notation introduced in Section 4.2. Table 4 shows the intermediate values used for a number of such expressions.

A further variation on this notation also allows us to specify a local semantics for expressions like *the first Tuesday* in temporal expressions like *the first Tuesday in July*, or like *the last year in the last year of the millenium*; see Table 5. To produce final interpretations of these, the interpreter has to construct the set of elements that correspond to the head noun (for example, a list of the ISO dates that correspond to the Tuesdays in a given month), and then select the *n*th element from that set.

5 Handling Embedded Constructions

The TIMEX specification allows for the embedding of one TIMEX within another. Consider an example like the following:

String	Representation
the first Tuesday	1D2
the second day	2D
the last Tuesday	\$D2
the last day	\$D

Table 5: Ordinally-specified Elements

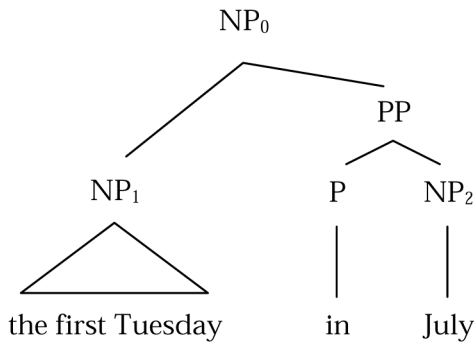


Figure 1: The syntactic structure of an embedded TIMEX

(9) `<TIMEX2>the first Tuesday
in<TIMEX2>July</TIMEX2></TIMEX2>`

The bulk of the embedded TIMEXs provided as examples in the TIMEX guidelines are, like this one, of the form [NP PP], where the head NP contains a TIMEX, and the PP contains another TIMEX that is modified by the head NP. Syntactically, these structures are of the form shown in Figure 1.

For our purposes, it is convenient to first think of these structures as consisting of three, rather than two, TIMEXs, corresponding to the three subscripted NP nodes in this tree. The outermost TIMEX, corresponding to NP₀, is the one whose value we are ultimately interested in; this is computed by combining the semantics of the two constituent TIMEXs, corresponding to NP₁ and NP₂, and the preposition indicates how this combination should be carried out.

Structurally, the recognizer may first determine that there are two separate TIMEXs here:

(10) `<TIMEX2>the first
Tuesday</TIMEX2> in
<TIMEX2>July</TIMEX2>`

Each of these TIMEXs can be given the appropriate local semantics by the recognizer; the recognizer then reorganizes this structure to mirror the embedding required by the TIMEX guidelines, to produce the structure shown in example (10) above; effectively, NP₁ disappears as a distinct constituent, and its intermediate semantics are inherited by NP₀.

We then leave it to the interpreter to combine the intermediate semantics of NP₀ with the intermedi-

ate semantics of NP₂ to produce a final semantics for NP₀: schematically, we have

$$(11) \quad NP_0(\text{VAL}) = NP_0(\text{T-VAL}) * NP_2(\text{T-VAL})$$

where ‘*’ is the combinatory operation that corresponds to the preposition used. The operation required is specified by the recognizer as the value of the temporary attribute T-REL, which represents the semantics of the preposition.

The following three examples demonstrate a variety of possibilities, showing both the intermediate (T-VAL) and final (VAL) semantic interpretations in each case:

(12) `<TIMEX2 VAL="1999" T-VAL="$Y"
T-REL="OF">the last year of
<TIMEX2 VAL="1" T-VAL="+0">this
millennium</TIMEX2></TIMEX2>`

(13) `<TIMEX2 VAL="1998-01-31"
T-VAL="$D" T-REL="OF">the last
day of <TIMEX2 VAL="1998-01"
T-VAL="xxxx-01">January
</TIMEX2></TIMEX2>`

(14) `<TIMEX2 VAL="1998-01-31"
T-VAL="$D" T-REL="OF">the last
day of <TIMEX2 VAL="1998-01"
T-VAL="1998-01">January
1998</TIMEX2></TIMEX2>`

Note that, when the embedded TIMEX is fully specified, as in the last example here, it would be possible for the recognizer to calculate the final value of the whole expression; however, for consistency we leave this task to the interpreter.

The semantics of the indicated T-REL depend on the types of its arguments. In the cases above, for example, the operation is one of selecting an ordinal-specified element of a list; but where the entity is a period rather than a point, as in *the first six months of 2005*, the operation is one of delimiting the period in question.

Of course, other forms of embedding are possible. In appositions, the syntactic structure can be thought of as [NP NP]; as in the case of embedded PPs, the TIMEX representation effectively promotes the semantics of the first NP to be the semantics of the whole. Again, we show both VAL and T-VAL values here, and the relevant T-REL.

(15) `<TIMEX2 VAL="1998-12-29"
T-VAL="xxxx-xx-xx"`

```
T-REL="EQUAL">my birthday,  
<TIMEX2 VAL="1998-12-29"  
T-VAL="xxxx-12-29">December  
twenty-ninth</TIMEX2></TIMEX2>
```

- (16) <TIMEX2 VAL="196" T-VAL="196"
T-REL="EQUAL">the 1960s, <TIMEX2
VAL="196" T-VAL="PXD">the days of
free love</TIMEX2></TIMEX2>

Here, the fact that the T-REL is EQUAL causes the interpreter to combine the values of the two TIMEXs, with points taking precedence over durations.

6 Conclusions

In this paper, we have argued that, in the context of interpreting temporal expressions, there is value in identifying a level of semantic representation that corresponds to the meaning of these expressions outside of any particular document context. This idea is not in itself new, and many existing systems appear to make use of such representations. However, we have proposed that this level of representation be made explicit; and by providing an encoding of this level of representation that is an extension of the existing TIMEX2 annotations in terms of element attributes and their values, we make it possible to assess the performance of systems with respect to intermediate values, final values, or both, using standard evaluation tools.

We have developed the representation described here on the basis of the set of 265 examples provided in the TIMEX2 guidelines (Ferro et al., 2005), and this set of annotated examples is available to the community.⁶ The approach described here is implemented in DANTE, a text processing system which produces normalised values for all TIMEXs found in a document. The recognition component of the system, which constructs the intermediate representations described here, is implemented via just over 200 rules written in the JAPE language:⁷ time expressions are thus recognised using finite state patterns, but we then apply a syntactic check, using the Connexor parser, to ensure that we have identified the full extent of each temporal expression, appropriately extending the extent when this is not the case.

⁶See www.clt.mq.edu.au/timex.

⁷JAPE is provided as part of the GATE tools (Cunningham et al., 2002).

We are currently testing this representation and its means of derivation against the data from the 2004 TERN competition. Our results are broadly comparable to those achieved by other systems (for example, Chronos or TempEx), though they can not be compared directly since the reported evaluations at the TERN competition use data which are not public and therefore not available to us.

7 Acknowledgements

We acknowledge the support of the 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*.
- 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.
- J. R. Hobbs and F. Pan. 2004. An ontology of time for the semantic web. *ACM Transactions on Asian Language Information Processing*, 3(1):66–85, March.
- I. Mani, J. Pustejovsky, and R. Gaizauskas, editors. 2005. *The Language of Time*. Oxford Univ. Press.
- 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.
- I. Pratt and N. Francez. 2001. Temporal prepositions and temporal generalized quantifiers. *Linguistics and Philosophy*, 24:187–222.
- J. Pustejovsky, J. Castaño, R. Ingria, R. Gaizauskas, R. Saur, A. Setzer, and G. Katz. 2003. TimeML: Robust Specification of Event and Temporal Expressions in Text. In *IWCS-5, Fifth International Workshop on Computational Semantics*.
- E. Saquete, P. Martínez-Barco, and R. Muñoz. 2002. Recognising and Tagging Temporal Expressions in Spanish. In *Proc. of LREC'02: Workshop on Annotation Standards for Temporal Information in Natural Language, Las Palmas, Spain*.
- F. Schilder. 2004. Extracting meaning from temporal nouns and temporal prepositions. *ACM Transactions on Asian Language Information Processing*, 3(1):33–50, March.