An Annotation Scheme for Reichenbach's Verbal Tense Structure

Leon Derczynski and **Robert Gaizauskas**
Department of Computer Science
University of Sheffield, UK
{leon,robertg}@dcs.shef.ac.uk

Abstract

In this paper we present RTMML, a markup language for the tenses of verbs and temporal relations between verbs. There is a richness to tense in language that is not fully captured by existing temporal annotation schemata. Following Reichenbach we present an analysis of tense in terms of abstract time points, with the aim of supporting automated processing of tense and temporal relations in language. This allows for precise reasoning about tense in documents, and the deduction of temporal relations between the times and verbal events in a discourse. We define the syntax of RTMML, and demonstrate the markup in a range of situations.

1 Introduction

In his 1947 account, Reichenbach offered an analysis of the tenses of verbs, in terms of abstract time points. Reichenbach details nine tenses (see Table 1). The tenses detailed by Reichenbach are past, present or future, and may take a simple, anterior or posterior form. In English, these apply to single verbs and to verbal groups (e.g. *will have run*, where the main verb is *run*).

To describe a tense, Reichenbach introduces three abstract time points. Firstly, there is the speech time, S. This represents the point at which the verb is uttered or written. Secondly, event time E is the time that the event introduced by the verb occurs. Thirdly, there is reference time R; this is an abstract point, from which events are viewed. In Example 1, speech time S is when the author created the discourse (or perhaps when the reader interpreted it). Reference time R is *then* – an abstract point, before speech time, but after the event time E, which is the leaving of the building. In this sentence, one views events from a point in time later than they occurred.

(1) *By then, she had left the building.*

While we have rich annotation languages for time in discourse, such as TimeML[1] and TCNL[2], none can mark the time points in this model, or the relations between them. Though some may provide a means for identifying speech and event times in specific situations, there is nothing similar for reference times. All three points from Reichenbach's model are sometimes necessary to calculate the information used in these rich annotation languages; for example, they can help determine the nature of a temporal relation, or a calendrical reference for a time. We will illustrate this with two brief examples.

(2) *By April 26th, it was all over.*

In Example 2, there is an anaphoric temporal expression describing a date. The expression is ambiguous because we cannot position it absolutely without an agreed calendar and a particular year. This type of temporal expression is interpreted not with respect to speech time, but with respect to reference time (Ahn et al., 2005). Without a time frame for the sentence (presumably provided earlier in the discourse), we cannot determine which year the date is in. If we are able to set bounds for R in this case, the time in Example 2 will be the April 26th adjacent to or contained in R; as the word *by* is used, we know that the time is the April 26th following R, and can normalise the temporal expression, associating it with a time on an absolute scale.

Temporal link labelling is the classification of relations between events or times. We might say an event of *the airport closed* occurred **after** another event of *the aeroplane landed*; in this case, we have specified the type of temporal relation between two events. This task is difficult to automate (Verhagen et al., 2010). There are clues in

---

[1] http://www.timeml.org; Boguraev et al. (2005).
[2] See Han et al. (2006).

discourse that human readers use to temporally relate events or times. One of these clues is tense. For example:

(3) *John told me the news, but I had already sent the letter.*

Example 3 shows a sentence with two verb events – *told* and *had sent*. Using Reichenbach's model, these share their speech time S (the time of the sentence's creation) and reference time R, but have different event times. In the first verb, reference and event time have the same position. In the second, viewed from when John told the news, the letter sending had already happened – that is, event time is before reference time. As reference time R is the same throughout the sentence, we know that the letter was sent before John mentioned the news. Describing S, E and R for verbs in a discourse and linking these points with each other (and with times) is the only way to ensure correct normalisation of all anaphoric and deictic temporal expressions, as well as enabling high-accuracy labelling of some temporal links.

Some existing temporal expression normalisation systems heuristically approximate reference time. GUTime (Mani and Wilson, 2000) interprets the reference point as "the time currently being talked about", defaulting to document creation date. Over 10% of errors in this system were directly attributed to having an incorrect reference time, and correctly tracking reference time is the only way to resolve them. TEA (Han et al., 2006) approximates reference time with the most recent time temporally before the expression being evaluated, excluding noun-modifying temporal expressions; this heuristic yields improved performance in TEA when enabled, showing that modelling reference time helps normalisation. HeidelTime (Strötgen and Gertz, 2010) uses a similar approach to TEA but does not exclude noun-modifying expressions.

The recently created WikiWars corpus of TIMEX2 annotated text prompted the comment that there is a "need to develop sophisticated methods for temporal focus tracking if we are to extend current time-stamping technologies" (Mazur and Dale, 2010). Resources that explicitly annotate reference time will be direct contributions to the completion of this task.

Elson and McKeown (2010) describe how to relate events based on a "perspective" which is calculated from the reference and event times of an event pair. They construct a natural language generation system that requires accurate reference times in order to correctly write stories. Portet et al. (2009) also found reference point management critical to medical summary generation.

These observations suggest that the ability to automatically determine reference time for verbal expressions is useful for a number of computational language processing tasks. Our work in this area – in which we propose an annotation scheme including reference time – is a first step in this direction.

In Section 2 we describe some crucial points of Reichenbach's model and the requirements of an annotation schema for tense in natural language. We also show how to reason about speech, event and reference times. Then, in Section 3, we present an overview of our markup. In Section 4 we give examples of annotated text (fictional prose and newswire text that we already have another temporal annotation for), event ordering and temporal expression normalisation. Finally we conclude in Section 5 and discuss future work.

2 Exploring Reichenbach's model

Each tensed verb can be described with three points; speech time, event time and reference time. We refer to these as S, E and R respectively. Speech time is when the verb is uttered. Event time is when the action described by the verb occurs. Reference time is a viewpoint from where the event is perceived. A summary of the relative positions of these points is given in Table 1.

While each tensed verb involves a speech, event and reference time, multiple verbs may share one or more of these points. For example, all narrative in a news article usually has the same speech time (that of document creation). Further, two events linked by a temporal conjunction (e.g. *after*) are very likely to share the same reference time.

From Table 1, we can see that conventionally English only distinguishes six tenses. Therefore, some English tenses will suggest more than one arrangement of S, E and R. Reichenbach's tense names suffer from this ambiguity too, but to a much lesser degree. When following Reichenbach's tense names, it is the case that for past tenses, R always occurs before S; in the future, R is always after S; and in the present, S and R are simultaneous. Further, "anterior" suggests E

Relation	Reichenbach's Tense Name	English Tense Name	Example
E<R<S	Anterior past	Past perfect	*I had slept*
E=R<S	Simple past	Simple past	*I slept*
R<E<S	Posterior past		*I expected that ..*
R<S=E			*I would sleep*
R<S<E			
E<S=R	Anterior present	Present perfect	*I have slept*
S=R=E	Simple present	Simple present	*I sleep*
S=R<E	Posterior present	Simple future	*I will sleep (Je vais dormir)*
S<E<R	Anterior future	Future perfect	*I will have slept*
S=E<R			
E<S<R			
S<R=E	Simple future	Simple future	*I will sleep (Je dormirai)*
S<R<E	Posterior future		*I shall be going to sleep*

Table 1: Reichenbach's tenses; from Mani et al. (2005)

before R, "simple" that R and E are simultaneous, and "posterior" that E is after R. The flexibility of this model permits the full set of available tenses (Song and Cohen, 1988), and this is sufficient to account for the observed tenses in many languages.

Our goal is to define an annotation that can describe S, E and R (speech, event and reference time) throughout a discourse. The lexical entities that these times are attached to are verbal events and temporal expressions. Therefore, our annotation needs to locate these entities in discourse, and make the associated time points available.

2.1 Special properties of the reference point

The reference point R has two special uses. When sentences or clauses are combined, grammatical rules require tenses to be adjusted. These rules operate in such a way that the reference point is the same in all cases in the sequence. Reichenbach names this principle **permanence of the reference point**.

Secondly, when temporal expressions (such as a TimeML TIMEX3 of type DATE, but not DURATION) occur in the same clause as a verbal event, the temporal expression does not (as one might expect) specify event time E, but instead is used to position reference time R. This principle is named **positional use of the reference point**.

2.2 Context and the time points

In the linear order that events and times occur in discourse, speech and reference points persist until changed by a new event or time. That is, the reference time from one sentence will roll over to the next sentence, until it is repositioned explicitly by a tensed verb or time. To cater for subordinate clauses in cases such as reported speech, we add a caveat – S and R persist as a discourse is read in textual order, for each context. We can define a context as an environment in which events occur, such as the main body of the document, reported speech, or the conditional world of an *if* clause (Hornstein, 1990). For example:

(4) *Emmanuel had said "This will explode!", but changed his mind.*

Here, *said* and *changed* share speech and reference points. Emmanuel's statement occurs in a separate context, which the opening quote instantiates, ended by the closing quote (unless we continue his reported speech later), and begins with an S that occurs at the same time as *said*'s E. This persistence must be explicitly stated in RTMML.

2.3 Capturing the time points with TimeML

TimeML is a rich, developed standard for temporal annotation. There exist valuable resources annotated with TimeML that have withstood significant scrutiny. However TimeML does not address the issue of annotating Reichenbach's tense model with the goal of understanding reference time or creating resources that enable detailed examination of the links between verbal events in discourse.

Although TimeML permits the annotation of tense for <EVENT>s, it is not possible to unambiguously map its tenses to Reichenbach's model. This restricts how well we can reason about verbal

events using TimeML-annotated documents. Of the usable information for mapping TimeML annotations to Reichenbach's time points, TimeML's `tense` attribute describes the relation between S and E, and its `aspect` attribute can distinguish between `PERFECTIVE` and `NONE` – that is, between $E < R$ and a conflated class of $(E = R) \lor (R < E)$. Cases where $R < E$ are often awkward in English (as in Table 1), and may even lack a distinct syntax; the French *Je vais dormir* and *Je dormirai* both have the same TimeML representation and both translate to *I will sleep* in English, despite having different time point arrangements.

It is not possible to describe or build relations to reference points at all in TimeML. It may be possible to derive the information about S, E and R directly represented in our scheme from a TimeML annotation, though there are cases – especially outside of English – where it is not possible to capture the full nuance of Reichenbach's model using TimeML. An RTMML annotation permits simple reasoning about reference time, and assist the labelling of temporal links between verb events in cases where TimeML's tense and aspect annotation is insufficient. This is why we propose an annotation, and not a technique for deriving S, E, and R from TimeML.

3 Overview of RTMML

The annotation schema RTMML is intended to describe the verbal event structure detailed in Reichenbach (1947), in order to permit the relative temporal positioning of reference, event, and speech times. A simple approach is to define a markup that only describes the information that we are interested in, and can be integrated with TimeML. For expositional clarity we use our own tags but it is possible (with minor modifications) to integrate them with TimeML as an extension to the standard.

Our procedure is as follows. Mark all times and verbal events (e.g. TimeML TIMEX3s and those EVENTs whose lexical realisation is a verb) in a discourse, as $T_1..T_n$ and $V_1..V_n$ respectively. We mark times in order to resolve positional uses of the reference point. For each verbal event V_i, we may describe or assign three time points S_i, E_i, and R_i. Further, we will relate T, S, E and R points using disjunctions of the operators $<$, $=$ and $>$. It is not necessary to define a unique set of these points for each verb – in fact, linking them

across a discourse helps us temporally order events and track reference time. We can also define a "discourse creation time," and call this S_D.

(5) *John said, "Yes, we have left".*

If we let *said* be V_1 and *left* be V_2:

- $S_1 = S_D$

From the tense of V_1 (simple past), we can say:

- $R_1 = S_1$
- $E_1 < R_1$

As V_2 is reported speech, it is true that:

- $S_2 = E_1$

Further, as V_2 is anterior present:

- $R_2 = S_2$
- $E_2 < R_2$

As the $=$ and $<$ relations are transitive, we can deduce an event ordering $E_2 < E_1$.

3.1 Annotation schema

The annotation language we propose is called RTMML, for Reichenbach Tense Model Markup Language. We use standoff annotation. This keeps the text uncluttered, in the spirit of *ISO LAF* and *ISO SemAF-Time*. Annotations reference tokens by index in the text, as can be seen in the examples below. Token indices begin from zero. We explicitly state the segmentation plan with the `<seg>` element, as described in Lee and Romary (2010) and *ISO DIS 24614-1 WordSeg-1*.

The general speech time of a document is defined with the `<doc>` element, which has one or two attributes: an ID, and (optionally) `@time`. The latter may have a normalised value, formatted according to TIMEX3 (Boguraev et al., 2005) or TIDES (Ferro et al., 2005), or simply be the string `now`.

Each `<verb>` element describes a tensed verbal group in a discourse. The `@target` attribute references token offsets; it has the form `target="#token0"` or `target="#range(#token7,#token10)"` for a 4-token sequence. Comma-separated lists of offsets are valid, for situations where verb groups are non-contiguous. Every `verb` has a unique value in its `@id` attribute. The tense of a verb group is described using the attributes `@view`

Relation name	Interpretation
POSITIONS	$T_a = R_b$
SAME_TIMEFRAME	$R_a = R_b[, R_c, ..R_x]$
REPORTS	$E_a = S_b$

Table 2: The meaning of a certain link type between verbs or times a and b.

(with values *simple*, *anterior* or *posterior*) and @tense (*past*, *present* or *future*).

The `<verb>` element has optional @s, @e and @r attributes; these are used for directly linking a verb's speech, event or reference time to a time point specified elsewhere in the annotation. One can reference document creation time with a value of `doc` or a temporal expression with its id (for example, t1). To reference the speech, event or reference time of other verbs, we use hash references to the event followed by a dot and then the character s, e or r; e.g., v1's reference time is referred to as #v1.r.

As every tensed verb always has exactly one S, E and R, and these points do not hold specific values or have a position on an absolute scale, we do not attempt to directly annotate them or place them on an absolute scale. One might think that the relations should be expressed in XML links; however this requires reifying time points when the information is stored in the relations between time points, so we focus on the relations between these points for each `<verb>`. To capture these internal relations (as opposed to relations between the S, E and R of different verbs), we use the attributes se, er and sr. These attributes take a value that is a disjunction of $<$, $=$ and $>$.

Time-referring expressions are annotated using the `<timerefx>` element. This has an @id attribute with a unique value, and a @target, as well as an optional @value which works in the same was as the `<doc>` element's @time attribute.

```
<rtmml>
Yesterday, John ate well.
 <seg type="token" />
 <doc time="now" />
 <timerefx xml:id="t1" target="
      #token0" />
 <verb xml:id="v1" target="#token3"
      view="simple" tense="past"
      sr=">" er="=" se=">"
      r="t1" s="doc" />
</rtmml>
```

In this example, we have defined a time *Yesterday* as t1 and a verbal event *ate* as v1. We have categorised the tense of v1 within Reichenbach's nomenclature, using the `verb` element's @view and @tense attributes.

Next, we directly describe the reference point of v1, as being the same as the time t1. Finally, we say that this verb is uttered at the same time as the whole discourse – that is, $S_{v1} = S_D$. In RTMML, if the speech time of a verb is not otherwise defined (directly or indirectly) then it is S_D. In cases of multiple voices with distinct speech times, if a speech time is not defined elsewhere, a new one may be instantiated with a string label; we recommend the formatting *s*, *e* or *r* followed by the verb's ID.

This sentence includes a positional use of the reference point, annotated in v1 when we say r="t1". To simplify the annotation task, and to verbosely capture a use of the reference point, RTMML permits an alternative annotation with the `<rtmlink>` element. This element takes as arguments a relation and a set of times and/or verbs. Possible relation types are POSITIONS, SAME_TIMEFRAME (annotating permanence of the reference point) and REPORTS for reported speech; the meanings of these are given in Table 2. In the above markup, we could replace the `<verb>` element with the following:

```
<verb xml:id="v1" target="#token3"
      view="simple" tense="past"
      sr=">" er="=" se=">" s="doc" />
<rtmlink xml:id="l1" type="POSITIONS">
  <link source="#t1" />
  <link target="#v1" />
</rtmlink>
```

When more than two entities are listed as targets, the relation is taken as being between an optional `source` entity and each of the `target` entities. Moving inter-verbal links to the `<rtmlink>` element helps fulfil *TEI p5* and the *LAF* requirements that referencing and content structures are separated. Use of the `<rtmlink>` element is not compulsory, as not all instances of positional use or permanence of the reference point can be annotated using it; Reichenbach's original account gives an example in German.

3.2 Reasoning and inference rules

Our three relations $<$, $=$ and $>$ are all transitive. A minimal annotation is acceptable. The S, E and R points of all verbs, S_D and all Ts can represent nodes on a graph, connected by edges labelled

with the relation between nodes.

To position all times in a document with maximal accuracy, that is, to label as many edges in such a graph as possible, one can generate a closure by means of deducing relations. An agenda-based algorithm is suitable for this, such as the one given in Setzer et al. (2005).

3.3 Integration with TimeML

To use RTMML as an ISO-TimeML extension, we recommend that instead of annotating and referring to `<timerefx>`s, one refers to `<TIMEX3>` elements using their `tid` attribute; references to `<doc>` will instead refer to a `<TIMEX3>` that describes document creation time. The attributes of `<verb>` elements (except `xml:id` and `target`) may be be added to `<MAKEINSTANCE>` or `<EVENT>` elements, and `<rtmlink>`s will refer to event or event instance IDs.

4 Examples

In this section we will give developed examples of the RTMML notation, and show how it can be used to order events and position events on an external temporal scale.

4.1 Annotation example

Here we demonstrate RTMML annotation of two short pieces of text.

4.1.1 Fiction

From *David Copperfield* by Charles Dickens:

(6) *When he had put up his things for the night he took out his flute, and blew at it, until I almost thought he would gradually blow his whole being into the large hole at the top, and ooze away at the keys.*

We give RTMML for the first five verbal events from Example 6 RTMML in Figure 1. The fifth, `v5`, exists in a context that is instantiated by `v4`; its reference time is defined as such. We can use one `link` element to show that `v2`, `v3` and `v4` all use the same reference time as `v1`. The temporal relation between event times of `v1` and `v2` can be inferred from their shared reference time and their tenses; that is, given that `v1` is anterior past and `v2` simple past, we know $E_{v1} < R_{v1}$ and $E_{v2} = R_{v2}$. As our `<rtmlink>` states $R_{v1} = R_{v2}$, then $E_{v1} < E_{v2}$. Finally, `v5` and

`v6` happen in the same context, described with a second SAME_TIMEFRAME link.

4.1.2 Editorial news

From an editorial piece in TimeBank (Pustejovsky et al., 2003) (AP900815-0044.tml):

(7) *Saddam appeared to accept a border demarcation treaty he had rejected in peace talks following the August 1988 cease-fire of the eight-year war with Iran.*

```
<doc time="1990-08-15T00:44" />
<!-- appeared -->
<verb xml:id="v1" target="#token1"
  view="simple" tense="past" />
<!-- had rejected -->
<verb xml:id="v2"
  target="#range(#token9,#token10)"
  view="anterior" tense="past" />
<rtmlink xml:id="l1"
  type="SAME_TIMEFRAME">
  <link target="#v1" />
  <link target="#v2" />
</rtmlink>
```

Here, we relate the simple past verb *appeared* with the anterior past (past perfect) verb *had rejected*, permitting the inference that the first verb occurs temporally after the second. The corresponding TimeML (edited for conciseness) is:

```
Saddam <EVENT eid="e74" class="I_STATE">
appeared</EVENT> to accept a border
demarcation treaty he had <EVENT eid="e77"
class="OCCURRENCE">rejected</EVENT>

<MAKEINSTANCE eventID="e74" eiid="ei1568"
 tense="PAST" aspect="NONE" polarity="POS"
 pos="VERB"/>
<MAKEINSTANCE eventID="e77" eiid="ei1571"
 tense="PAST" aspect="PERFECTIVE"
 polarity="POS" pos="VERB"/>
```

In this example, we can see that the TimeML annotation includes the same information, but a significant amount of other annotation detail is present, cluttering the information we are trying to see. Further, these two `<EVENT>` elements are not directly linked, requiring transitive closure of the network described in a later set of `<TLINK>` elements, which are omitted here for brevity.

4.2 Linking events to calendrical references

RTMML makes it possible to precisely describe the nature of links between verbal events and times, via positional use of the reference point. We will link an event to a temporal expression, and suggest a calendrical reference for that expression, allowing the events to be placed on a calendar. Consider the below text, from wsj_0533.tml in TimeBank.

```
<doc time="1850" mod="BEFORE" />          view="simple" tense="past" />         type="SAME_TIMEFRAME">
<!-- had put -->                          <!-- would gradually blow -->            <link target="#v1" />
<verb xml:id="v1"                         <verb xml:id="v5"                        <link target="#v2" />
  target="#range(#token2,#token3)"          target="#range(#token26,#token28)"     <link target="#v3" />
  view="anterior" tense="past" />           view="posterior" tense="past"          <link target="#v4" />
<!-- took -->                               se="=" er=">" sr=">"                   </rtmlink>
<verb xml:id="v2" target="#token11"        r="#v4.e" />                           <rtmlink xml:id="l2"
  view="simple" tense="past" />            <!-- ooze -->                             type="SAME_TIMEFRAME">
<!-- blew -->                              <verb xml:id="v6"                         <link target="#v5" />
<verb xml:id="v3" target="#token17"         target="#range(#token26,#token28)"      <link target="#v6" />
  view="simple" tense="past" />             view="posterior" tense="past"         </rtmlink>
<!-- thought -->                            se="=" er=">" sr=">" />
<verb xml:id="v4" target="#token24"        <rtmlink xml:id="l1"
```

Figure 1: RTMML for a passage from David Copperfield.

(8) *At the close of business Thursday, 5,745,188 shares of Connaught and C$44.3 million face amount of debentures, convertible into 1,826,596 common shares, had been tendered to its offer.*

```
<doc time="1989-10-30" />
<!-- close of business Thursday -->
<timerefx xml:id="t1"
  target="#range(#token2,#token5)" />
<!-- had been tendered -->
<verb xml:id="v1"
  target="#range(#token25,#token27)"
  view="anterior" tense="past" />
<rtmlink xml:id="l1" target="#t1 #v1">
  <link target="#t1" />
  <link target="#v1" />
</rtmlink>
```

This shows that the reference time of `v1` is `t1`. As `v1` is anterior, we know that the event mentioned occurred before *close of business Thursday*. Normalisation is not a task that RTMML addresses, but there are existing methods for deciding which Thursday is being referenced given the document creation date (Mazur and Dale, 2008); a time of day for *close of business* may be found in a gazetteer.

4.3 Comments on annotation

As can be seen in Table 1, there is not a one-to-one mapping from English tenses to the nine specified by Reichenbach. In some annotation cases, it is possible to see how to resolve such ambiguities. Even if view and tense are not clearly determinable, it is possible to define relations between S, E and R; for example, for arrangements corresponding to the simple future, $S < E$. In cases where ambiguities cannot be resolved, one may annotate a disjunction of relation types; in this example, we might say "$S < R$ or $S = R$" with `sr="<="`.

Contexts seem to have a shared speech time, and the $S - R$ relationship seems to be the same throughout a context. Sentences which contravene this (e.g. *"By the time I ran, John will have arrived"*) are rather awkward.

RTMML annotation is not bound to a particular language. As long as a segmentation scheme (e.g. WordSeg-1) is agreed and there is a compatible system of tense and aspect, the model can be applied and an annotation created.

5 Conclusion and Future Development

Being able to recognise and represent reference time in discourse can help in disambiguating temporal reference, determining temporal relations between events and in generating appropriately tensed utterances. A first step in creating computational tools to do this is to develop an annotation schema for recording the relevant temporal information in discourse. To this end we have presented RTMML, our annotation for Reichenbach's model of tense in natural language.

We do not intend to compete with existing languages that are well-equipped to annotate temporal information in documents; RTMML may be integrated with TimeML. What is novel in RTMML is the ability to capture the abstract parts of tense in language. We can now annotate Reichenbach's time points in a document and then process them, for example, to observe interactions between temporal expressions and events, or to track reference time through discourse. This is not directly possible with existing annotation languages.

There are some extensions to Reichenbach's model of the tenses of verbs, which RTMML does not yet cater for. These include the introduction of a reference interval, as opposed to a reference point, from Dowty (1979), and Comrie's suggestion of a second reference point in some circumstances (Comrie, 1985). RTMML should cater for these extensions.

Further, we have preliminary annotation tools and have begun to create a corpus of annotated texts that are also in TimeML corpora. This will allow a direct evaluation of how well TimeML can represent Reichenbach's time points and their relations. To make use of Reichenbach's model in automatic annotation, given a corpus, we would like to apply machine learning techniques to the RTMML annotation task. Work in this direction should enable us to label temporal links and to anchor time expressions with complete accuracy where other systems have not succeeded.

6 Acknowledgements

References

D. Ahn, S.F. Adafre, and MD Rijke. 2005. Towards task-based temporal extraction and recognition. In *Dagstuhl Seminar Proceedings*, volume 5151.

B. Boguraev, J. Castano, R. Gaizauskas, B. Ingria, G. Katz, B. Knippen, J. Littman, I. Mani, J. Pustejovsky, A. Sanfilippo, et al. 2005. TimeML 1.2. 1: A Formal Specification Language for Events and Temporal Expressions.

B. Comrie. 1985. *Tense*. Cambridge University Press.

D.R. Dowty. 1979. *Word meaning and Montague grammar*. Kluwer.

D. Elson and K. McKeown. 2010. Tense and Aspect Assignment in Narrative Discourse. In *Proceedings of the Sixth International Conference in Natural Language Generation*.

L. Ferro, L. Gerber, I. Mani, B. Sundheim, and G. Wilson. 2005. TIDES 2005 standard for the annotation of temporal expressions. Technical report, MITRE.

International Organization for Standardization. 2009a. *ISO DIS 24612 LRM - Language Annotation Framework (LAF)*. ISO/TC 37/SC 4/WG 2.

International Organization for Standardization. 2009b. *ISO DIS 24614-1 LRM - Word Segmentation of Text - Part 1: Basic Concepts and General Principles (WordSeg-1)*. ISO/TC 37/SC 4/WG 2.

International Organization for Standardization. 2009c. *ISO DIS 24617-1 LRM - Semantic Annotation Framework - Part 1: Time and Events (SemAF-Time)*. ISO/TC 37/SC 4/WG 2.

B. Han, D. Gates, and L. Levin. 2006. From language to time: A temporal expression anchorer. In *Temporal Representation and Reasoning (TIME)*, pages 196–203.

N. Hornstein. 1990. *As time goes by: Tense and universal grammar*. MIT Press.

K. Lee and L. Romary. 2010. Towards Interoperability of ISO Standards for Language Resource Management. In *International Conference on Global Interoperability for Language Resources*.

I. Mani and G. Wilson. 2000. Robust temporal processing of news. In *Proceedings of the 38th Annual Meeting on ACL*, pages 69–76. ACL.

I. Mani, J. Pustejovsky, and R. Gaizauskas. 2005. *The language of time: a reader*. Oxford University Press, USA.

P. Mazur and R. Dale. 2008. What's the date?: high accuracy interpretation of weekday names. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 553–560. ACL.

P. Mazur and R. Dale. 2010. WikiWars: A New Corpus for Research on Temporal Expressions. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 913–922. ACL.

F. Portet, E. Reiter, A. Gatt, J. Hunter, S. Sripada, Y. Freer, and C. Sykes. 2009. Automatic generation of textual summaries from neonatal intensive care data. *Artificial Intelligence*, 173(7-8):789–816.

J. Pustejovsky, P. Hanks, et al. 2003. The Time-Bank corpus. In *Corpus Linguistics*, volume 2003, page 40.

H. Reichenbach. 1947. The Tenses of Verbs. *Elements of Symbolic Logic*, pages 287–98.

A. Setzer, R. Gaizauskas, and M. Hepple. 2005. The role of inference in the temporal annotation and analysis of text. *Language Resources and Evaluation*, 39(2):243–265.

F. Song and R. Cohen. 1988. The interpretation of temporal relations in narrative. In *Proceedings of the 7th National Conference of AAAI*.

J. Strötgen and M. Gertz. 2010. HeidelTime: High quality rule-based extraction and normalization of temporal expressions. In *Proceedings of the 5th Workshop on Semantic Evaluation*, pages 321–324. ACL.

M. Verhagen, R. Saurí, T. Caselli, and J. Pustejovsky. 2010. SemEval-2010 task 13: TempEval-2. In *Proceedings of the 5th Workshop on Semantic Evaluation*, pages 57–62. ACL.