Chapter 6
Using a Framework of Tense and Aspect

> For years I have endeavored to break through the veil which shrouded it, and at last the time came when I seized my thread and followed it.
>
> *The Final Problem*
> SIR ARTHUR CONAN DOYLE

6.1 Introduction

This chapter investigates a linguistic framework for tense and aspect. Analysis of the temporal relation typing problem in Chap. 4 suggested two directions for investigation. Temporal signals were one of these; tense and aspectual differences were the second prevalent category. Having investigated temporal signals in Chap. 5, this chapter is dedicated to the other major source of temporal ordering information in difficult links.

Tense and aspect are used to describe temporal aspects of events which are expressed with verbs. It is intuitive that tense and aspect will be of some value for determining the type of temporal relation that holds between two verb events, and evidence in human-annotated corpora supports this intuition.

Event-event relations are the hardest to label (Chap. 4). Around 45 % of links in TempEval (a temporal annotation evaluation exercise, see Sect. 3.4.4.4) event-event tasks cannot reliably be labelled automatically (see Sect. 4.2.2). Further, verb-verb links make up a significant amount of the difficult links identified in Sect. 4.2.

Relations involving at least one argument with tense or aspect information are prevalent. They are also difficult to label. Verb-verb links make up around a third of TimeBank's TLINKs, and tensed verb-verb links the largest share of that set, so of all verb-verb relations, the majority are between two tensed verbs.

Ordering time expressions and events in the same sentence is a also somewhat difficult task. In TimeBank, almost half of all TLINKs are between a time and event. Of these, half are between an event and timex in the same sentence, where the timex is a date or time.

L.R.A. Derczynski, *Automatically Ordering Events and Times in Text*,
Studies in Computational Intelligence 677, DOI 10.1007/978-3-319-47241-6_6

Table 6.1 Frequency of TimeML tense and aspect on verb events in TimeBank

Tense	Aspect	Count
PAST	NONE	1975
PRESENT	NONE	803
INFINITIVE	NONE	762
PRESPART	NONE	360
PRESENT	PERFECTIVE	270
FUTURE	NONE	262
PRESENT	PROGRESSIVE	162
PASTPART	NONE	150
PAST	PERFECTIVE	88
NONE	PERFECTIVE	20
PAST	PROGRESSIVE	19
PRESENT	PERFECTIVE_PROGRESSIVE	17
FUTURE	PROGRESSIVE	5
FUTURE	PERFECTIVE	4
NONE	PROGRESSIVE	3
NONE	PERFECTIVE_PROGRESSIVE	2
PASTPART	PERFECTIVE	2
PAST	PERFECTIVE_PROGRESSIVE	1
PRESPART	PERFECTIVE	1

Data-driven approaches to the relation typing task are hampered in two ways. Firstly, there is a shortage of ground truth training data, which is in turn partially due to the high cost of annotation. As [1] point out, this leads to low volumes of instances for many combinations of tense and aspect values for pairs of events (see Table 6.1), potentially hampering automatic hypothesis learning. Secondly, the variation of expression annotatable using TimeML is relatively limited, describing three "tenses"[1] (past and past participle, present and present participle, and future) and three "aspects" (none, perfective and progressive). This markup language may be insufficiently descriptive to capture the relations implied by all the variations in linguistic use of tense and aspect.

Reichenbach [2] offers a theoretical framework for analysis of tense and aspect that can be used to predict constraints on temporal orderings between verb events based on their tense and aspect, and also between times and tensed verbs. Applying Reichenbach's framework requires tense and aspect information, which is provided in TimeML (meaning that it might be possible to apply this framework without a major annotation effort).

[1]In TimeML v1.2, the tense attribute of events has values that are conflated with verb form. This conflation is deprecated in versions of TimeML more recent than that in which TimeBank is annotated.

Application of the framework gives a partial idea of the temporal ordering between a suitable pair of events or an event and timex (except durations and sets). These rough orderings can be used to constrain of the set of possible TimeML relation types for any given pair. For example, a suggestion of “overlap” constrains possible TimeML relations to “simultaneous/includes/included_by”.

It may be the case that machine learning methods are unable to make effective use of the tense information available in TimeBank. Phenomena such as tense shifts between events have been shown to help humans temporal ordering [3], and therefore may convey some temporal information. However, the percentage of links with tense shifts is roughly the same in the general case (40 % in TimeBank) and the difficult link set (36 %). As these figures are roughly the same, it may be that supervised approaches fail to make generalisations that take advantage of the information given in tense shifts.

Prior work has gone some way to determining the utility of tense in the relation typing task. The USFD system in TempEval-2007 [4] found that the supplied tense was not a helpful feature for event-timex linking (though aspect was), though that it did provide some benefit to event-event ordering when the events were in the same or adjacent sentences.

Reichenbach’s framework may offer a method for determining or approximating temporal orderings over this significant part of the difficult link set (and also in the general case). In this chapter, we offer a full account of Reichenbach’s framework in the context of TimeML, and investigate how consistent the framework is with gold-standard temporally annotated data, before offering methods for integrating it into a temporal relation typing approach.

The rest of this chapter is structured as follows. Firstly, we discuss in abstract terms a conceptual model for time. Second, there is an introduction to Reichenbach’s framework and a description of how it interacts with temporal expressions as well as verb events, followed by a summary of related work. Next, validation of the framework is attempted by describing how the framework can be related to TimeML and then an evaluation of it against ground truth temporal relation type information. The framework’s relation type constraints are then applied to the temporal relation typing task alongside data from TimeML annotations, as part of a machine learning approach to relation typing, and results presented. It is found that Reichenbach framework is potentially helpful. To allow inclusion of what the framework provides that is not in TimeML already, an annotation scheme for the framework is introduced (RTMML) which may also be used as an extension to TimeML. Finally, the chapter concludes with a discussion of applications of the framework and future work.

6.2 Timelines in Language

Time, as experienced and expressed by humans, seems to be linear. Events begin and end at points along this line, through which travel is always unidirectional; each event’s end can come no earlier than its beginning.

Time is often described using the same language as space, as touched upon in Sect. 5.4.4. We talk about *time travel*, use words such as *faster*, *before* and *at* and specify directions such as *forward* and *backward*. The linguistic relation between expression of time and space is sometimes taken to extremes; some have suggested that we travel through time facing backwards, because we can only see the past and not the future [5]. The spatio/temporal polysemy is even learned by classifier models when attempting to detect temporal usages of words (Sect. 5.6.4.3). This linguistic similarity is rooted in the way that humans understand non-literal motion (such as in temporal transitions) using the same cognitive resources as we understand literal (e.g. spatial) motion [6].

Given that time is a linear and effectively continuous [7] dimension which progresses unidirectionally [8] but can be conceived of in either direction [9], we talk about its description in language with a model of time as uni-dimensional (cf. McTaggart's A-series [10]).

As a line is a conceptually simple spatial representation of a single linear dimension (such as time), we shall describe our temporal dimension by means of a "**timeline**". We are constantly at a point that we refer to as the present. This point exists on the timeline as a separator between the past and the future. Our timeline can thus be described as three non-overlapping parts: past, present and future.

The time at which an utterance is heard or read is always the present. Some way is required of referring to events at points on a timeline that happen any time but the perceiver's present. One can perhaps define a method of absolute description of positions on a timeline, maybe by use of a calendar or clock[2] to determine origin locations. However, the attachment to every event of a label defined using an external scale causes event descriptions to be awkward both to write and to read (even ignoring the overhead of temporal scale creation, maintenance and reference). A potentially simpler mechanism is to describe events relative to each other; one may like to talk of things happening either at present, in the part of the timeline before it, or the part coming later.

These three parts correspond directly to the rudiments of tense in language; the past tense, present tense and future tense permit expression of events within the past, at the present, or within the future part of a timeline (cf. McTaggart's B-series). Thus, simple tense usage allows positioning of events within regions on a timeline relative to the present; and so, in that it describes temporally relative points, tense is inherently deictic [11, 12]. The tenses corresponding to these three categories are known as **absolute tenses**.

Given such a tense structure, one may identify two temporal points upon the timeline. One is the time at which the description of the event is uttered or perceived, and the other, that may be in any of the three timeline parts, corresponds to the time

[2] In fact, each of these "absolute references" eventually relies upon events. A year is the event of a full cycle of the earth around the sun, and a second is the duration of a certain number of caesium isotope decay events. The common era calendar is centred around an agreed point based on a described event; each day's start (e.g. midnight) is determined by the event of a specific angle of rotation of the earth upon its axis relative to the sun.

that the described action took place. This simple structure allows us to temporally express events relative to the present.

However, the ability to relate events to each other – critical to planning and story-telling – is still difficult with this system. If we are to mention an event and then express another event in terms of that (e.g. *The race will be over and I will have won*), one must be able to treat the first event as a sort of basis or origin for positioning the second. In this example, the *winning* happens in one of the three parts of a timeline where the "present" is at or after the race's completion. To express this, we need what amounts to double-deixis; there is one three-part structuring of the timeline where the present centres upon the time of utterance, and another with the present situated around the race's completion.

In language, this double-deixis can be accounted for in a system of tense and aspect. It is required not only to describe a primary event relative to its primary deixis, but also then to describe a secondary event relative to the primary event. This might involve a relocation of the listener such that the secondary event's temporal position is described in terms that they are familiar with – such as the 3-part past/present/future model – centred not upon the listener's present, but instead around the primary event described. In our example, the *winning* is described not relative to the time the sentence is uttered, but in terms of the event of the race's end.

As well as recognising divisions of past, present and future, we can describe this secondary structuring of a timeline around an event by use of anterior, simple and past tenses. These correspond to events described before, at or after the initially-described event. Continuing to use the race example, the race is over at some point in the future, and the *winning* happens before this – anterior to the primary event. As the primary event occurs in the future, we say that *I will have won* is in the *anterior future* tense. This gives us a tense system that allows the description both of events relative to now, and also of events relative to each other that is also readily describable using a timeline.

It is worth noting at this point that, being irrealis from the point of reference, the future tense is often considered a modality rather than a tense – certainly in English. This is echoed by McTaggart's argument for incoherence of the A-series (the absolute, external, ordered sequence of events) [10]; he essentially claims that time is incoherent, as we know that events have an innate ordering, so how could we not see what that ordering is? Any given event, as time advances, will be past and will have been future. Jaszczolt details with this another way, by putting forward that temporality is modal, with different tenses (or other representations of time) having varying degrees of certainty [13]. Both of these arguments hinge on the future being modal. In any event, one generally needs linguistic devices with which to describe the future, and tense is such a device, where the future is just one partition (the others being past and present).

6.3 Description of the Framework

The core of the framework comprises three abstract time points – speech time, event time and reference time – which are related to each other in terms of equality (e.g. simultaneity), precedence or succession. The tense and aspect of verbs are then described using these points, which we introduce properly next. Finally, interactions between verbs are formalised in terms of relations between the abstract time points of each verb. This section introduces the basic framework as proposed by Reichenbach, and then discusses its limitations and puts forward additional proposals for extending the framework.

6.3.1 Time Points

To describe a tense, Reichenbach introduces three abstract time points. Firstly, there is the speech time,[3] 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. Klein [15] describes it as "the time to which a claim is constrained".

In Example 18, speech time S is when the author created the discourse (or perhaps when the reader interpreted it).

Example 18 By then, she had left the building.

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. Therefore, the final configuration is $E < R < S$.

6.3.2 Reichenbachian Tenses

Reichenbach details nine tenses (see Table 6.2). 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 non-infinitive verbs and to verbal groups consisting of head verb and auxiliaries. The tense system describes abstract time points for each tensed verb and how they may interact, both for a single verb and with other events described by verbs.

In Reichenbach's view, different tenses specify different relations between E, R and S. Table 6.2 shows the six tenses conventionally distinguished in English. As

[3] For this book, speech time is equivalent to DCT, unless otherwise explicitly positioned by discourse. Under Fillmore's description [14], this is the same as always setting speech time *S* equal to encoding time *ET* and not decoding time *DT*.

Table 6.2 Reichenbach's tenses; from [16]

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 R < S = E R < S < E	Posterior past		I expected that I would sleep
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 S = E < R E < S < R	Anterior future	Future perfect	I will have slept
S<R = E	Simple future	Simple future	I will sleep (Je dormirai)
S<R<E	Posterior future		I shall be going to sleep

there are more than six possible ordering arrangements of S, E and R, some English tenses might suggest more than one arrangement. Reichenbach's named tenses names also suffer from this ambiguity when converted to $S/E/R$ structures, albeit to a 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 before R, "simple" that R and E are simultaneous, and "posterior" that E is after R. The flexibility of this framework is sufficient to allow it to account for a very wide set of tenses, including all those described by [17], and this is sufficient to account for the observed tenses in many languages. Past, present and future tenses imply $R < S$, $R = S$ and $S < R$ respectively. Anterior, simple and posterior tenses imply $E < R$, $E = R$ and $R < E$ respectively.

6.3.3 Verb Interactions

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* - see Chap. 5) are very likely to share the same reference time. Basic methods of linking between verb events or linking verbs to fixed points on a time scale are described below.

6.3.3.1 Special Properties of the Reference Point

The reference point *R* has two special uses. These relate to verbs in the same *temporal context* (see Sect. 6.3.4 below) and to the effect of time expressions on verbs.

Permanence

Firstly, when sentences are combined to form a compound sentence, tensed mean verbs interact, and implicit 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**; *"We can interpret these rules as the principle that, although the events referred to in the clauses may occupy different time points, the reference point should be the same for all clauses"*. Figure 6.1 contains an example of this principle.

Positional

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**.

In Example 19, an explicit time (*10 o'clock*) determines our reference point through positional use.

Example 19 It was 10 o'clock, and Sarah had brushed her teeth.

The verb group *had brushed* is anterior past tense; that is, $E < R < S$. The event is complete before the reference time – that is, at any point until *10 o'clock* – and so the relation between the event and timex can be determined (*brushed* BEFORE *10 o'clock*).

6.3.3.2 Example Reichenbachian Verb-Verb Links

All three points from Reichenbach's framework are sometimes necessary to position an event on a timeline or in relation to another event. For example, they can help determine the nature of a temporal relation, or a calendar reference for a time. We illustrate this two brief examples.

Example 20 In February 1917, the Germans landed their offensive. By April 26th*, it was all over.*

Example 20 shows a temporal expression describing a day – April 26^{th}. The expression is ambiguous because we cannot position it absolutely without knowing

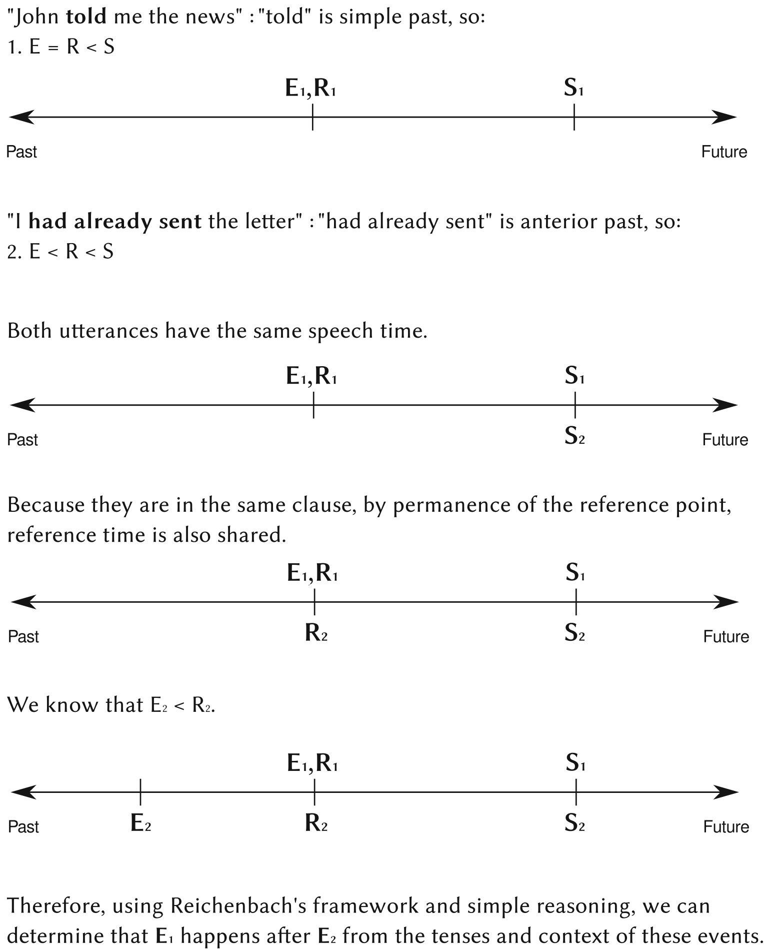

Fig. 6.1 An example of permanence of the reference point

which year it refers to. This type of temporal expression is interpreted with respect to reference time, not with respect to speech time [18]. 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 20 will be the April 26^{th} adjacent to or contained in R; as the word *by* is used, we know that the time is the April 26^{th} following R, and can normalise the temporal expression, associating it with a time on an absolute scale.

Example 21 John told me the news, but I had already sent the letter.

Example 21 and Fig. 6.1 show a sentence with two verb events – *told* and *had sent*. Using Reichenbach's framework, 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.

Example 22 contains a more advanced example. It shows a pair of temporally related verbs taken from the list of difficult links found earlier (see Sect. 4.3.1).

Example 22 A committee of outside directors for the Garden City, N.Y., unit is $\underline{\textit{evaluating}}_{e1}$ *the proposal ; the parent* $\underline{\textit{asked}}_{e2}$ *it to respond by Oct. 31.*

One can determine the temporal relation between events `e1` and `e2` from the tenses in this sentence without particularly complex reasoning. In the example, `e1` is present progressive, and `e2` is past tense. Te end point of *evaluating* (`e1`) is after the end of `e2` and after the time of the example's writing. We can also see that the end of `e2` is in the past – the *asked* started and finished before document creation time (DCT), and certainly finished before *evaluating* finishes. This tense-based reasoning gives a constrained set of temporal relation types.

6.3.4 Temporal Context

In the linear order that events and times are introduced in discourse, speech and reference points persist until changed by a new event or time. Observations during the course of this work suggest that 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 make discussion of sets of verbs with common reference times easy, we call each of these groups a **temporal context**.

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 temporal context. A context is an environment in which events occur, and may be the main body of the document, a tract of reported speech, or the conditional world of an *if* clause [19]. For example:

Example 23 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* – or, to be precise, *said*'s event time E_{said}.

Temporal contexts may be observed frequently in natural language discourse. For example, the main body of a typical news article shares the same reference point, reporting other events and speech as excursions from this context. Each conditional world of events invoked by an "if" statement will share the same context. Events or times linked with a temporal signal will share a reference point, and thus be explicitly placed into the same temporal context.

As described in Chap. 4 of [19] in his description of the sequence of tenses with regard to Reichenbach's framework, permanence of the reference point does not apply between main events and embedded phrases, relative clauses or quoted speech. These occur within a separate temporal context, and it is likely that they will have their own reference time (and possibly even speech time, for example, in the case of quoted speech). In order to apply permanence of the reference point, it ought only be applied within the same temporal context. Verbs to which permanence may be applied are said by Reichenbach to be those to which the grammatical rules of the **sequence of tenses** (an abstract set of grammatical rules not described in his paper) apply. Different contexts will have a consistent reference point, and so permanence of the reference point may be applied to verbs within that context in order to gain information about their temporal relations. Permanence does not apply across different temporal contexts.

Dowty [20] hints at the concept of temporal context with the idea of the **temporal discourse interpretation principle** (TDIP). This states:

> Given a sequence of sentences S_1, S_2, …, S_n to be interpreted as a narrative discourse, the reference time of each sentence S_i (for i such that $1 < i - n$) is interpreted to be:
>
> (a) a time consistent with the definite time adverbials in S_i, if there are any;
>
> (b) otherwise, a time which immediately follows the reference time of the previous sentence S_{i-1}.

The TDIP accounts for a set of sentences which share a reference and speech point. However, as with other definitions of temporal context, this principle involves components that are difficult to automatically determine (e.g. "consistent with definite time adverbials"). Miller et al. [21] may offer a parallel account of temporal context, in their definition of narrative containers, though it is down to empirical comparison to answer this question.

As discussed above, Temporal context describes the events which may temporally linked using Reichenbach's framework in order to helpfully constrain the set of temporal relations between each pair. It is therefore useful to automatic relation typing approaches to know the bounds of each temporal context. However, this information is not present in TimeML annotations and not readily available from discourse. This gives the problem of having to model temporal context, in order to decide which event verb-event verb TLINKs to apply the framework.

Modeling temporal context requires the grouping of tensed verb event pairs so that only those in which both events are in the same temporal context are together. Simple techniques for achieving this could work on sentence proximity. In Time-Bank, there are 1 167 event-event TLINKs where both arguments are tensed verbs, of which 600 are in the same sentence and a further 313 are in adjacent sentences. Further techniques for temporal context modelling are detailed in experiments below. Proximity alone may not be sufficient, given this chapter's earlier observations about quoted speech, re-positioning of the reference point and so on; however, it is a simple starting point.

While positional use of the reference point indicates a new (or change to an established) temporal context, and permanence of the reference point can only persist within the same temporal context, the principle of quoted speech (above) permits linking across some temporal contexts.

6.3.5 Quoted Speech

The framework can also be used to described adjustment of speech, reference and event time around reported, quoted speech. Although not mentioned in Reichenbach's original account, the principle emerges directly from his framework, and is as follows. When a verb is used to initiate quoted, reported speech, the speech time for that quote is equivalent to the event time of the initiating verb.

Example 24 shows two verb events: one initiates quoted speech (*told*), and the other is within this reported speech (*hold*).

Example 24 This morning General Powell told reporters, "We will hold a press conference shortly."

In this case, the event time of *told* corresponds to the speech time of *hold*. This form of reasoning allows us to connect events within quoted speech to those outside it. It may be referred to as **positional use of the speech point**. Just as with positional use of the reference point, where another entity determines how the *reference point* should be interpreted, positional use of the speech point occurs when another entity (in this case an event) determines how the *speech point* should be interpreted.

Exposition of the principle benefits from [19]'s modestly extended definition of speech time, as follows:

> The key to the analysis is the recognition that the S point has two related yet logically distinct properties: (i) it is a deictic anchor and (ii) it has a default interpretation in which it is mapped onto the utterance time if not otherwise interpreted.
>
> Distinguishing these two properties of the S point permits the formulation of a sequence of tense rule for embedded finite clauses. In this case, the rule associates an embedded point, S_{n-1}, with a higher point, E_n.

6.3.6 Limitations of the Framework

This section contains a discussion of some shortcomings of Reichenbach's tense framework and – where relevant – the proposed solutions.

6.3.6.1 Limited Tenses

The included tenses and aspects are insufficiently expressive to cover the gamut of linguistic expressions of temporality. One may look at lexical semantic models of tense and aspect in English to discover a wider inventory of possible tenses and aspects in that language [22], or examine other languages with richer aspect systems to see what the framework glosses over in those cases (e.g. [23]). Limitations of the Reichenbachian perfect can be seen from Table 6.2, where there is more than one triple that corresponds to the future perfect. Nevertheless, many tense and aspect systems can be described in terms of Reichenbach's framework, albeit not always as a 1:1 mapping.

6.3.6.2 Progressive Aspect

The progressive is used for events that have both a start and end and are currently ongoing; that is, in-progress activities. This makes it possible to refer to points within an event. However, Reichenbach's framework is point-based, and point-based temporal algebras generally assume that when point events are referenced, they are only referenced in terms of being before, after or simultaneous with another temporal entity. This makes it difficult to accurately represent more complex verbal event structures. Introducing interval reasoning to the framework can help (that is, dealing with intervals in terms of start and end points, instead of a single point for the whole), although it is sufficient to achieve this through treating events as a coupled start and end point (where the start is never after the end). This has the advantage of permitting semi-interval type reasoning (see Sect. 3.2.0.3). We discuss this further in Sect. 6.4.2

6.3.6.3 *On* Dates

Positional use of the reference point tells us that R is equivalent to a timex in the clause, if given. Because the algebra the framework uses to describe tenses is point-based, the start and end of the given time period are equal to the start and end of the reference time. This gives problems when a described event takes place during a provided timex, but does not have the same start and stop times. Example 25 is taken from [24]:

Example 25 Mary left England on May the 22nd, 1979

In this case, although Reichenbach's framework tells us that $R = E$ and that R is equivalent to *May the 22nd, 1979*, it is false that the leaving – E – took place simultaneously with the date; rather, it was a subpart of this 24 h interval. One solution to this unintuitive behaviour is to replace the reference point with a reference interval, having distinct start and end points if required.

6.3.6.4 Non-English Tense System

Some languages are difficult to accommodate in Reichenbach's framework. To accommodate Russian, for example, one must make specific and extensive additions to the framework, including binary temporal relations between points for each verb [25]. Such a system can be extended to cover a large range of Slavic languages [26], though is too complex to implement for a first attempt at automated temporal annotation using Reichenbach's framework.

Further, Reichenbach's framework is less useful given a language that has a limited tense system. It relies on a richness of expression placed in verb tenses. Without this richness, the value of applying the framework is reduced. For example, Chinese does not inflect verbs to express tense, but rather uses grammatical constructions, particles and temporal adverbials to describe time. The system is still somewhat less complex (regarding Reichenbach's framework) than that of English or French. The habitual, present, present progressive and stative can all be expressed the same way.

Example 26 我吃吗 (wǒ chī mǎ) – "I eat horse"

A simple sentence is given in Example 26. This can be interpreted in English as *"I prefer to eat horse"*, *"I am currently eating horse"* or *"I will eat horse"*, *"I ate horse"*; contextual markets are required for clarification. The default interpretation is that of simple present tense. Past tense can be signified with guō (过), and completion with le (了), both of are placed directly after the verb. It is therefore possible to capture the relation between speech and event points, and we can determine if the reference point is after the event or not. There is nothing to clarify the difference between simple and anterior tenses, and (as in English) the simple present is also used to indicate habitual truths (e.g. *I eat horse*). However, unlike English, the simple present progressive (e.g. *I am eating horse*) looks identical to the habitual use. Further information is expressed through temporal adverbials and not considered tense. The general lack of inflection or cohesive verb groups suggests that Reichenbach's framework can only be applied to Chinese in a limited fashion, decreasing its general utility.

6.3.6.5 Split Reference Point

Some tensed temporal descriptions of events are difficult to framework with just a single reference point. For example, from [27]:

Example 27

- *"I shall have been going to see John."* (that is, there is some point in the past at which I anticipated seeing John; note this is not a description of habitual behaviour)
- $S < R_1 < E < R_2$

It is true that the tenses and abstract points provided by the three-point framework are insufficient to capture this statement, without invoking an extra verb event.

However, in TimeBank no such contrived utterances were found during candid examinations or error analysis from applying the framework to predict TimeML relations.

6.3.6.6 Reification of the Reference Point

Tanaka [28] takes exception to the abstract nature of the reference point, and that it is never reified or explicitly lexicalised. He questions the requirement for reference time in a system of tense, and raises a few examples that are difficult to express using Reichenbach's framework. Tanaka's criticism and example are as follows.

Example 28

- Now Megumi will marry Kazuhiko next month.
- $S < E = R$

In Example 28, the temporal adverbial *next month* is used to position the reference point, R. With the tense used here – simple future – this also places E (the time of marrying) during *next month*, which is the correct interpretation. However, Tanaka suggests that the framework does not explain the influence of *Now* in this sentence; for which verbs does it fix the reference point? This criticism could be viewed as a variation on the requirement for two reference points to describe some verbs.

We can, in fact, provide a concrete solution in this case. One could attach *Now* to the auxiliary verb *will*, which provides a correct arrangement of points under Reichenbach's framework and is also an effective way of representing the situation in TimeML. It is not proposed that this is a satisfactory solution in terms of linguistic theory, rather, that it is a solution in computational for the purpose of automatically determining the nature of a given temporal relation.

6.4 Validating the Framework Against TimeBank

Having described Reichenbach's framework of tense and aspect and introduced related linguistic and temporal concepts, we now investigate how the framework compares with real data. Before applying Reichenbach's framework to the TimeML relation typing task, it is important to check if it is descriptively adequate. As it is possible to identify a set of candidate links where the argument types are of the right type (tensed verb events), the relation types of these can be compared with those suggested by the framework.

In order to evaluate its suggestions, temporal relation types suggested by the framework can be compared with a human-annotated ground truth, such as TimeBank. The framework can be applied to TLINKs where both arguments are tensed verbs, given tense and aspect information. This fits the difficult case identified in Chap. 4, that of event-event links involving some shift of tense. When ordering events based on positional use or permanence of the reference point, the set of TLINKs is further constrained to those where both arguments are in the same temporal context.

To compare the framework with TimeML-annotated resources, a number of decisions must be taken as part of an interpretation of the framework. Firstly, the Reichenbachian tense and aspect attributes do not directly match those in TimeML; some kind of mapping needs to be created between these two tense/aspect systems. One must convert a tense from TimeML into an arrangement of speech, event and reference point. Reichenbach suggests nine "basic" tenses and his system allows many arrangements of these points; TimeML separates tense and aspect and allows for values quite different to those included in Reichenbach's framework.

Secondly, Reichenbach is vague about temporal context. It is unclear from TimeML annotations alone which sets of verbs can be considered to be in the same "temporal context" (see Sect. 6.3.4). Reichenbach simply states that the framework is intended to follow the sequence of verbs. The descriptions of the "sequence of tenses" suggest it is difficult to implement programatically with current technology (see e.g. Chap. 4 of [19]), and require accurate identification of reported speech, embedded phrases, relative clauses, reference-time shifting temporal adverbials and so on. This presents a number of complex syntactic and linguistic scoping tasks that may be difficult to perform automatically. Therefore, one needs an approximation of temporal context in order to choose which verb pairs to attempt to relate.

Aside from these two decisions which help determine which event pairs to link and how to represent them, it is useful to construct a table describing temporal relation constraint according to the framework. The suggested type of relation between two events (or an event and a timex) – given their tense and aspect in Reichenbach's framework, and that permanence of the reference points holds between them – is not provided elsewhere, and some kind of relation matrix needs to be determined. To use tense and aspect values for temporal relation typing within the framework, we are concerned with possible arrangements of two event times given two verbs that represent these events, and need to describe the relation between event times. This provides a means to extract useful ordering information even in the situation that reference times do not match perfectly.

In the two-event sentence of Example 29, *fished* is anterior present with arrangement $E < S_1 = R_1$ and *eat* is simple future, with arrangement $S < R_2 = E_2$.

Example 29 *"I have fished$_1$; John will eat$_2$."*

The event times are located such that *fished* wholly precedes *eat* with relation to the speech time, regardless of reference time's situation, leading to the equivalent of a TimeML BEFORE relation. It is not always possible to suggest a relation, perhaps due to a lack of information; for example, two events in the simple past cannot be temporally ordered relative to one another without further information (e.g. in "I went to school, you went to church").

Note that *eat*$_2$ could be interpreted as Reichenbachian posterior present, with arrangement $S = R_2 < E_2$. This gives the same temporal ordering of events, but through transitivity permits a shared reference point (i.e. $R_1 = R_2$). In this situation, as is sometimes the case in English, it is not possible to decide precisely which of posterior present and simple future applies. However, this is of little impact in this toy example when we are concerned primarily with determining relations between events; the reference point is only a means to that end.

To record relation types ready for later look-up, a two-dimensional matrix is constructed, with each axis labelled using all possible combinations of tense and aspect values under whatever scheme the first decision's outcome permits. Each cell in this matrix contains the temporal relation between event times suggested by the tenses and aspects of its axes.

The rule of permanence of the reference point could potentially be applied to a large number of temporal relations (e.g. those where both arguments are verb events), and if helpful, is the rule that could have the highest impact. For this reason, we only examine relations between two events where both events are verbs that have some tense information.

Below are details of a minimal interpretation and also an advanced interpretation of the framework, including quantitative assessment of their agreement with TimeBank's event annotations.

6.4.1 Minimal Interpretation of Reichenbach's Framework

The only criterion for permanence rule applicability not present in TimeML annotation is whether or not a pair of events are in the same temporal context. This was approximated by only considering event-event links where both events were in the same or adjacent sentences. In TimeML, event-event links between events inside or outside quotes and conditional/intentional constructs are annotated using other mechanisms, such as the SLINK, and not included in the relation typing task addressed. A selection of 211 links from TimeBank that match this approximation to temporal context were then manually examined to see if temporal context actually applied. Of this 211, a majority (146 – 69.2 %) had both arguments in the same context.

These cases were identified manually as follows. Firstly, the search space was narrowed to verb-verb events within the same or adjacent sentences. A random sample of these was drawn for manual examination. Instances where one event lay in a different temporal context were then excluded. A shift in reference time for the events means that they are not in the same context, and this was generally caused by a timex, one event being in an embedded phrase or relative clause, a special sense of a verb (such as habitual or stative), or one argument being in reported speech that the other is not.

To the 146 manually-annotated same-context temporal relations, temporal relation constraints derived from Reichenbach's framework were applied, to see if the gold standard annotated TimeML relation was consistent with the suggested constraints.

Reichenbach's framework can return some temporal ordering information for event pairs given a pair of tensed verb arguments in the same temporal context. As the only relations available are precedence and equality (simultaneity), the possible return values are: BEFORE, AFTER, OVERLAPPING (which subsumes simultaneous) and VAGUE. The relation VAGUE is assigned when, for example, both events occur before reference time but nothing else is known; this is not enough to describe any kind of order between events. These values are coarser than the TimeML relation types, and so the framework's output will serve to constrain available relation labels rather than describe a single one. For reference, BEFORE constrains the set to TimeML BEFORE or IBEFORE; AFTER to TimeML AFTER or IAFTER and OVERLAPPING to the remaining TimeML relations. An output of VAGUE offers no constraint at all.

TimeML's tense and aspect values were converted to Reichenbachian tenses using the schema given in Table 6.3. These Reichenbachian tenses were then used to find an R-E and an S-R ordering. These orderings for each verb were then coupled, assuming the *R* point for both verbs was shared, in order to determine an ordering between event times. Sometimes this was not possible (e.g. if both are simple past, while both can be described relative to the speech point, they cannot be described with any precision relative to the other); in this case, event orderings were made while falling back to assuming at least a shared *S* point. In other cases, sometimes only a vague relation was possible (e.g. if both are simple present, then they have both happened at some time – speech time – but we know nothing about their starts or ends relative to one another).

Table 6.4 details how constraints were selected. These constraints are translated to TimeML as follows:

Table 6.3 Minimal schema for mapping TimeML event tense and aspects to Reichenbach's framework

Tense	Non-perfect	Perfect
PAST	Simple past	Anterior past
PASTPART	Simple past	Anterior past
PRESENT	Simple present	Anterior present
PRESPART	Simple present	Anterior present
FUTURE	Simple future	Anterior future

Table 6.4 Event orderings based on the Reichenbachian tenses that are available in TimeML. Cell values describe the `e1` [*rel*] `e2` relationship. Note that TimeML has no unambiguous representation for anterior tenses, and so rows for these are not shown

e1 ↓; e2 →	Sim past	Pos past	Ant pres	Sim pres	Ant fut	Sim fut
Sim past	vague	after	vague	after	after	after
Pos past	before	vague	vague	vague	after	after
Ant pres	vague	vague	vague	after	vague	after
Sim pres	before	vague	vague	overlap	vague	after
Ant fut	before	before	vague	vague	vague	after
Sim fut	before	before	before	before	before	vague

Table 6.5 Accuracy of Reichenbach's framework with a subset of links manually annotated for being tensed verbs in the same temporal context

Output	Count	Consistent	% consistent
After	14	4	28.6 %
Overlap	19	15	84.2 %
Before	45	12	26.7 %
Total	78	31	39.7 %
Vague	68	–	–

- `before` - IBEFORE, BEFORE;
- `after` - IAFTER, AFTER;
- `overlap` - everything not covered by `before` or `after`;
- `vague` - no constraint.

As can be seen from prevalence of `vague` entries in the table, many combinations of tense offer no helpful constraint in terms of Allen's interval temporal relations. This is a hint that this particular interpretation of Reichenbach's tense may not see great performance increases when used for relation typing, and (depending on the actual distribution of tenses in the corpus) may not give a very clear picture of how accurate Reichenbach's model is.

The results are in Table 6.5. Indeed, it seems that, using this minimal interpretation, while in some cases Reichenbach's framework generates a temporal ordering that agrees with the TimeBank annotation, in the majority of situations the gold standard temporal orderings are inconsistent with what the framework interpretation suggests (i.e. the suggestion is wrong), or – almost half the time – the framework does not suggest anything useful (e.g. a "vague" response).

6.4.1.1 Minimal Interpretation Failure Analysis

Such low performance from a reasonable framework and interpretation demands analysis. Manual examination of the error set revealed many cases that Reichenbach's framework has problems with.

No Progressive

The framework doesn't handle the progressive aspect. If events have differing tenses (e.g. present and then future), the framework suggests by means of transitivity that the event time of the present-tensed verb is before that of the future-tensed verb. This makes this implicit assumption that the present-tensed item will have completed before the future-tensed item begins, ruling out any possibility of overlap. Progressive aspect is used as an indicator of ongoing processes, and could be used to weaken the constraint imposed by this minimal interpretation. For example, in *"I am running. Heston will cook."*, it is not certain that I will have finished running before the point that Heston starts cooking; that is to say, overlap is possible.

Poor Handling of Long-Running Events

The relations between S, E and R are over-specific information when discussing ongoing events. For example, in *"she hates us and always has hated us"*, a verb is described during another one, but there is a strong tense and aspect shift, from *hates* to *has hated*. Despite looking like a clear example of event ordering, the *hates* is a state that persists, and the speaker is just describing earlier points in the state's existence. However, this interpretation suggests that *hates* is simple present, $S = R = E$, and *has hated* is anterior present, $E < R = S$. This suggests that the event time of *hates* is after that of *has hated* when this is not actually the case. So, in this instance, Reichenbach's framework provides an over-specific response. Although an interpretation of *hates* as a proper interval immediately after the end of *has hated* is not impossible, it is somewhat tenuous, and the facts are too vaguely described to be as certain as the framework is.

Unusual Use of Tense

News presenters do unusual things with tense, and apply the reference point in a flexible manner. In *"And just last month, an off duty policeman is killed when a bomb explodes at another abortion clinic."* The meaning is clear, but the tenses do not compare well with a positional use of the reference point from the *last month* timex. The use of present tense suggests that the passive *killed* and the *explodes* events happen at the same time as the utterance. However, the present tense according to Reichenbach's framework suggests speech and reference time are equal, and in this case, the timex *last month* places speech time explicitly in the month previous to speech time – a direct conflict with the tense framework.

6.4.2 Advanced Interpretation of Reichenbach's Framework

The interpretation of Reichenbach's framework described above makes a few simplifications, and the results are poor. These simplifications may be the cause of incongruence between the framework's apparent suggestions and human-annotated ground-truth data. We improve the interpretation of Reichenbach's framework in the following ways, and re-check it. Some of this section's material also appears in [29].

Account of progressive aspect: In TimeML, aspect values are composed of two "flags", `perfective` and `progressive`, which may both be asserted on any tensed verb. Which Reichenbach's basic framework provides an account of the perfect (which TimeML calls perfective), it does not do the same for the progressive. This is resolved by splitting the event time E into start and finish points E_s and E_f between which the event obtains, as also done by e.g. [30]. For the simple tenses (where $R = E$), described as having TimeML aspect of NONE, it is assumed not that the event is a point, but that the event is an interval (just as in the progressive) and the reference time is *also* an interval, starting and finishing at the same times as the event (e.g. $R_s = E_s$ and $R_f = E_f$).

Variations of context assignment: Reichenbach's definition of which verbs may be linked through permanence of the reference point is a little vague, described as those that share a common reference point. This is approximated in a number of ways, results of each of which are presented: by considering all verb events in the same sentence; by considering all verb events in the same or an adjacent sentence; and by considering all verb events that have a common arrangement of both speech and reference time (e.g. all have the same arrangement of S and R). Ideally one should like to be able to track the speech and reference point through discourse, accounting for relative clauses, embedded phrases, reported speech and the like; in absence of a concerted investigation into performing these tasks reliably automatically, these approaches are approximations.

How to map TimeML to Reichenbach: Instead of the initial approach of mapping the TimeML tense and aspect values to a specific S/R/E point structure (e.g. a relative arrangement of speech, reference and event points) via one of the nine basic tenses specified in Reichenbach's framework, the TimeML tenses and aspects are mapped directly to S/R/E structures, using the translations shown in Table 6.6. For simplicity, PERFECTIVE_PROGRESSIVE aspect was converted to PERFECTIVE; the value makes up for 20 of 5974 verb events, or 0.34 % – a minority that should not have a great impact on overall results if altered slightly. One other simplification is that the participle "tenses" in TimeML (PASTPART and PRESPART) are interpreted in the same way as their non-participle equivalents, and so are not listed.

How to interpret relations suggested by the framework: Previously a label from one of four classes (before, after, overlap, vague) was assigned to a temporal relation, based on the tenses of its participant verb events. These classes did not accurately capture the 14 TimeML relations, and in many cases represented a disjunction of possible interval relation types. Working on the hypothesis that Reichenbach's framework may constrain a TimeML relation type to more than just four possible

Table 6.6 TimeML tense/aspect combinations, in terms of the Reichenbach framework

TimeML tense	TimeML aspect	Reichenbach structure
PAST	NONE	$E = R < S$
PAST	PROGRESSIVE	$E_s < R < S, R < E_f$
PAST	PERFECTIVE	$E_f < R < S$
PRESENT	NONE	$E = R = S$
PRESENT	PROGRESSIVE	$E_s < R = S < E_f$
PRESENT	PERFECTIVE	$E_f < R = S$
FUTURE	NONE	$S < R = E$
FUTURE	PROGRESSIVE	$S < R < E_f, E_s < R$
FUTURE	PERFECTIVE	$S < E_s < E_f < R$

Table 6.7 Example showing disjunctions of TimeML intervals applicable to describe the type of relation between A and B given their tense and aspect (e.g. to describe A rel B)

A ↓ B →	Perfect past	Present progressive
Perfect past	[any]	[before, ibefore, is_included, begins, during]
Present progressive	[after, iafter, includes, begun_by, during_inv]	[simultaneous, identity, during, during_inv, includes, is_included, ends, begins, ended_by, begun_by]

groupings, the table of tense-tense interactions is rebuilt, giving for each event pair a disjunction of TimeML relations instead of one of four labels. This has the advantage of adding distinctions that the minimal framework could not capture. Examples 30 and 31 would both be labeled "before" under that scheme, even though the latter is ambiguous regarding whether the progressive event has finished, and could signify an overlap.

Example 30 Anne had eaten breakfast. Bernard will sing.

Example 31 Chris was cleaning windows. Diana will sleep.

In this case, Example 30 suggests the TimeML relation *eaten* BEFORE *sing*, whereas because the end point of *cleaning* is not certain in Example 31, any of BEFORE, INCLUDES, or ENDED_BY may apply between *cleaning* and *sleep*. In this way, and with other arrangements of the speech, event and reference time, resolving relation types to disjunctions of potential interval relations provides a richer, more descriptive and more precise way of capturing the framework's output. An example is given in Table 6.7.

When constructing a table of potential TimeML TLINK `relType` values given two Reichenbachian tense structures with a disjunction of possible TimeML interval relation types in each cell, there is a finite set of combinations of relation types. That is to say, the disjunctions of interval relations indicated by various tense/aspect pair combinations frequently recur, and are not unique to each tense/aspect pair combination.

This finite set of interval relation disjunctions overlaps with the relation types grouped by Freksa (Sect. 3.2.3). For example, for two events E_1 and E_2, if the tense arrangement suggests that E_1 starts before E_2 (for example, E_1 is simple past and E_2 simple future), the available relation types for E_1/E_2 are BEFORE, IBEFORE, DURING, ENDED_BY and INCLUDES.

To clarify, given that $E_{1s} < E_{2s}$, and $E_s < E_f$ for any proper interval event (e.g. its start is before its finish), the arrangement of E_1 and E_2's finish points is left unspecified. The disjunction of possible interval relation types is as follows:

Table 6.8 Freksa semi-interval relations; adapted from Freksa (1992)

Relation	**Illustration**
X is *older* than Y Y is *younger* than X	`XXX????` `    YY`
X is *head to head* with Y	`XXX??` `YYYY`
X *survives* Y Y is *survived by* X	`????XXX` ` YY`
X is *tail to tail* with Y	`??XXX` ` YYYY`
X *precedes* Y Y *succeeds* X	`XXX?` `    YYY`
X is a *contemporary* of Y	` ?XXX???` `???YYY?`
X is *born before death* of Y Y *dies after birth* of X	` XXX?????` `?????YYY`

- $E_{1f} < E_{2s}$: before;
- $E_{1f} = E_{2s}$: ibefore;
- $E_{1f} > E_{2s}$, $E_{1f} < E_{2f}$: during;
- $E_{1f} = E_{2f}$: ended_by;
- $E_{1f} > E_{2f}$: includes.

In each case, these disjunctions correspond to the Freksa semi-interval relation E_1 YOUNGER E_2. As these Freksa semi-interval relations can be defined in terms of certain groups of Allen relations, the TimeML relations are almost equivalent to the Allen relations and the disjunctions of relations match these TimeML groups perfectly, the "output" of the Reichenbach framework regarding permanence of the reference point is given in Freksa semi-interval relations. The relations are shown in Table 6.8 and the TimeML tense/aspect interaction in Table 6.9.

Results

Interpreted in this way, Reichenbach's framework is more consistent with TimeBank than the earlier, minimal interpretation, generally supporting the framework's suggestions of event-event ordering among pairs of tensed verb events. Results are given in Table 6.10. In this table, an "accurate TLINK" is one where the relation type given in the ground truth is a member of the disjunction of relation types suggested by this interpretation of Reichenbach's framework.

Separate figures are provided for performance including and excluding cases where the disjunction of all link types (e.g. no constraint) is given. This is because achieving consistency with "no constraint" gives no information.

Table 6.9 TimeML tense/aspect pairs with the disjunction of TimeML relations they suggest, according to this chapter's enhanced interpretation of Reichenbach's framework

e1 ↓ e2 →	PAST-NONE	PAST-PROG	PAST-PERF	PRESENT-NONE	PRESENT-PROG	PRESENT-PERF	FUTURE-NONE	FUTURE-PROG	FUTURE-PERF
PAST-NONE	*all*	contemporary	succeeds	survivedby	survivedby	all	precedes	survivedby	before
PAST-PROGRESSIVE	contemporary	*contemporary*	survives	older	all	all	older	born before death	older
PAST-PERFECTIVE	precedes	survivedby	*all*	precedes	survivedby	precedes	before	survivedby	before
PRESENT-NONE	survives	younger	succeeds	*contemporary*	contemporary	survives	precedes	older	older
PRESENT-PROGRESSIVE	survives	all	survives	contemporary	*contemporary*	survives	older	born before death	older
PRESENT-PERFECTIVE	all	all	succeeds	survivedby	survivedby	*all*	before	survivedby	before
FUTURE-NONE	succeeds	younger	after	succeeds	younger	after	*all*	contemporary	survivedby
FUTURE-PROGRESSIVE	survives	dies after birth	survives	younger	dies after birth	survives	contemporary	*contemporary*	survives
FUTURE-PERFECTIVE	after	younger	after	younger	younger	after	survivedby	survivedby	*all*

Table 6.10 Consistency of temporal relation types suggested by Reichenbach's framework with ground-truth data. The non-all column refers to the number of incidences in which there was some kind of relation constraint, e.g., the framework did not give an unhelpful "all relation types possible" response

Context model	TLINKs	Accurate (%)	Non-"all"	Accurate (%)
None (all pairs)	1 167	81.5	481	55.1
Same sentence, same SR	300	88.0	95	62.1
Same sentence	600	71.2	346	50.0
Same/adjacent sentence, same SR	566	91.9	143	67.8
Same/adjacent sentence	913	78.3	422	53.1

Temporal context is complex to automatically detect, as detailed in Sect. 6.3.4 above. These results focus on the accuracy of the framework's temporal relation type constraints, given varying interpretations of temporal context.

The "same SR" context refers to modelling of temporal context as a situation where the ordering of reference and speech times remains constant (in terms of one preceding, occurring with or following the other). The rationale for this temporal context model is, because permanence of the reference point requires a shared reference time, for tenses to be meaningful in their context, the speech time must remain static. This simple same-ordering constraint on *S* and *R* does not preclude situations where speech or reference time move, but still remain in roughly the same order (e.g. if reference time moves from 9pm to 9.30pm when speech time is 3pm), which are in fact changes of temporal context (either because *R* is no longer shared or because *S* has moved).

In general, consistency is better than with the minimal interpretation discussed above. The "same SR" context gives good results, though has limited applicability in that it considers comparatively reduced sets of TLINKs (e.g. only half of same-sentence links). As both arguments having the same *S* and *R* occurs when they have the same TimeML tense, the only variant in these cases – in terms of data that contributes to Reichenbachian interpretation – is the TimeML aspect value. The increased "coverage" of the framework when given the constraint that TLINKs in which both arguments have the same TimeML tense hints that this is a critical factor in interpreting tense, and considering it may lead to improvements in temporal relation typing techniques that rely on aspect, such as that of [31]. The overall result is that Reichenbach's framework is capable of suggesting helpful relation types in some situations, and suggests further effort in applying and using the framework.

A slightly extended, standalone version of this validation can be found in [29].

6.5 Applying Reichenbach's Framework to Temporal Relation Typing

TimeML provides some of the information that Reichenbach's framework alone does not cater for. A combination of the two may lead to better labelling performance, but relying on Reichenbach's framework for rule-based temporal relation label constraint is insufficient. Application of the suggestions as integrated into a machine learning approach is discussed in the next section.

Reichenbach's framework for tense can be used to help determine the relation type between some times and events. This section describes use of the framework to develop features for enhancing temporal relation typing performance. These features are then added to the basic set defined in Sect. 4.4 as part of a temporal relation labelling classifier. The situations we examine are those where two verb events occur in the same temporal context, where a timex directly influences a verb event, and also verb events that report other verb events. A list of features is repeated below.

- text for each event;
- TimeML tense for each event;
- TimeML aspect for each event;
- modality for each event;
- cardinality for each event;
- polarity for each event;
- class for each event;
- part-of-speech for each event;
- are events in the same sentence?;
- are events in adjacent sentences?;
- do events have the same TimeML aspect?;
- do events have the same TimeML tense?;
- does event 1 textually precede event 2?

Because the framework relies on verb tense, all the situations described in this chapter can only work with events that are verbs and with time-referring expressions (that is, TIMEX3s of type DATE or TIME). It is therefore important to correctly determine the subset of all TLINKs that we try relation typing upon. Note that this subset selection is not the same as the relation identification task. The relation identification task requires, given a set of event and timex notifications, the selection of pairs that are temporally related. In contrast, for these experiments it is required, given a set of event, timex and TLINK annotations, to determine which of the TLINKs might benefit from the application of Reichenbach's framework. The relations covered are those that link same-context verbal events, that link events to times, and that link reporting events with events in reported speech. Throughout, the gold-standard EVENT and TIMEX3 annotations found in TimeBank are used, as well as the TLINKs identified there; the only task addressed is that of temporal relation typing.

6.5.1 Same Context Event-Event Links

The framework provides information for determining the ordering of events in the same temporal context (same context event-event links, or the SCEE dataset).

This situation applies to any two verb events that have a shared reference point. Verb events are identifiable by the event having a TimeML POS attribute of VERB, excluding those with a `tense` of `NONE` or `INFINITIVE`. A shared reference point is assumed for all verbs in the same sentence. Sentences are split using the Punkt sentence tokeniser for English [32]. These experiments use the minimal interpretation of Reichenbach's framework, described above.

One new feature is added to the standard feature set, corresponding to the relation type constraint suggested by our advanced interpretation of Reichenbach's framework (Sect. 6.4.2). The only ambiguity is over how to model temporal context. In this case, it is approached as being either event-event links with both arguments in the same sentence, or event-event links with both arguments in the same or adjacent sentences.

6.5.1.1 Results

The experiment was conducted with 10-fold cross validation, considering links from TimeBank v1.2, using relation type folding. The links within a document were never shared across a split (i.e., splits were made at document level). The experiments were conducted with relation folding (see Sect. 3.3.1). The impact of the new feature is measured by comparing classifier performance on SCEE links using the basic feature set and using the basic feature set plus the new feature. Features representing the text (i.e. lexical form) of events were removed as they consistently harmed performance, likely due to the sparsity of their values. Because the splits are determined randomly for cross-fold validation, every experiment is run three times and the mean performance figures given. The results are shown in Table 6.11, and a graph in Fig. 6.2. In this instance, the extended features provide a performance boost regardless of classifier choice.

Table 6.11 Using Reichenbach-suggested event ordering features representing permanence of the reference point, considering only same-sentence TLINKs. 562 examples

Classifier	Base features		Extended features	
	Accuracy (%)	Err. reduction (%)	Accuracy (%)	Err. reduction (%)
Baseline (MCC)	48.04	–	48.04	–
Maxent (megam)	57.47	22.86	**57.65**	23.19
Decision tree (ID3)	56.52	21.14	**57.47**	22.86
Naïve bayes	58.31	24.37	**58.72**	25.12

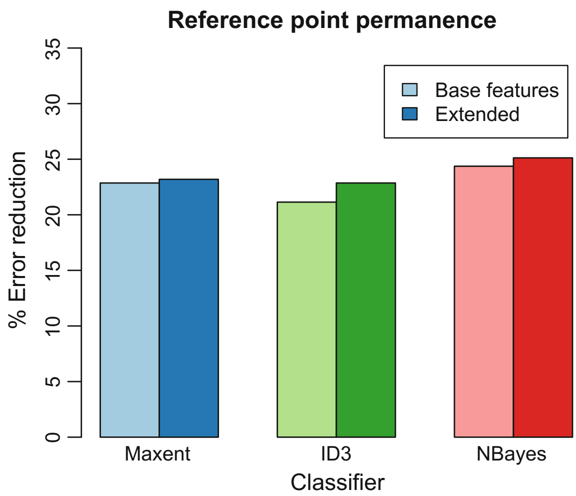

Fig. 6.2 Error reduction in SCEE links with and without features representing permanence of the reference point, modelling temporal context as same-sentence. The darker coloured columns correspond to error reduction using the feature derived from advanced interpretation of Reichenbach's framework

Table 6.12 Reichenbach-suggested event ordering feature representing permanence of the reference point. 858 examples

Classifier	Base features		Extended features	
	Accuracy (%)	Err. reduction (%)	Accuracy (%)	Err. reduction (%)
Baseline (MCC)	44.87	–	44.87	–
Maxent (megam)	62.28	31.58	**62.55**	32.07
Decision tree (ID3)	**59.21**	26.01	58.74	25.16
Naïve bayes	56.96	21.92	**57.58**	23.05

In the next case, the scope of temporal context is broadened to include cases where events are in adjacent sentences. Results are shown in Table 6.12. Here, the classifiers in which inductive bias tends toward the independence assumption do better with the extended feature set, but the decision tree does worse.

In both cases, there was a small performance increase from almost all classifiers with the introduction of the feature derived from advanced interpretation of Reichenbach's framework. Although the gains are not large, they are consistent.

Further work would concentrate on better discriminating which cases can be considered for application of permanence of the reference point. These are likely to span sentences. An annotation for delimiting these cases (e.g. temporal contexts) is put forward later, in Sect. 6.6.

6.5.2 *Same Context Event-Timex Links*

Reichenbach's framework provides explicit rules regarding the rôle of dates and times in respect to a verb within their temporal context (same context event-timex links: SCET). In these cases, the given time determines the time of the reference point, essentially reifying it (see Sect. 6.3.3).

To investigate whether constraints suggested by Reichenbach's framework can help in TLINK relation typing, we proceed as follows. For any verb event that is in the same sentence as a timex, if the timex modifies the event and the timex and event are linked through a TLINK, we assume that the timex positions the verb's reference point, and add a feature corresponding to this.

In all, 684 of the 6 418 available TLINKs could have this principle applied to them (10.7 % of all TLINKs). We are only interested in event-time links, of which there are 2 797; out of this set, 24.5 % (684) have event and time in the same sentence.

6.5.2.1 Features

One new feature is added to the base set (Sect. 4.4). As we are linking a timex and event under the assumption that there is a positional use of the reference point, the reference point is considered equivalent to the timex, and so the interesting temporal ordering is that between R and E. The reference point is determined using the advanced interpretation (Sect. 6.4.2, and the TimeML relation type between R and E constrained using Table 6.4 accordingly. In fact, as can be seen in Table 6.2, the type of tense embodies the E/R ordering: anterior tenses have $E < R$, simple tenses have $E = R$ and posterior tenses have $E > R$. Thus our symbolic label determining E/R relation (which is also E/T relation) assumes the value *anterior*, *simple* or *posterior*.

Dependency parses (generated by the Stanford Parser [33]) help determine whether or not a timex and event are syntactically connected. These parses also yield some extra information, which is included as features. These are:

- Direct modification: Does the timex directly modify the event? E.g., is the timex on the same dependency path as the event? (boolean);
- Temporal modification function: Is there a `tmod` relation in the dependency path from event to timex? (boolean);
- Final relation: The Stanford dependency relation of the timex node and its parent.

6.5.2.2 Results

Experiments were conducted with 10-fold document-level cross validation, using a folded relation set and no lexical features. Each experiment was run three times, and the mean result is reported (Fig. 6.3).

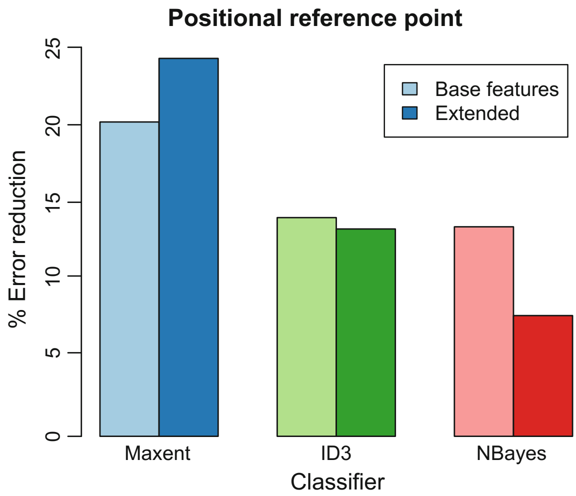

Fig. 6.3 Comparative performance on labelling event-time links where the time positions the reference point

Table 6.13 Performance when using dependency parse and Reichenbach-derived feature, in terms of relation typing accuracy and error reduction above the baseline. 684 instances

Classifier	Base features		Dep. features		RBach features		Dep. + RBach	
	Accuracy (%)	ER (%)	Accuracy (%)	ER (%)	Accuracy (%)	ER (%)	Accuracy (%)	ER (%)
Baseline (MCC)	66.67	–	66.67	–	66.67	–	66.67	–
Maxent (megam)	**73.39**	**20.18**	**74.71**	**24.12**	**74.75**	**24.24**	**74.76**	**24.26**
Decision tree (ID3)	71.35	14.04	70.03	10.09	71.05	13.16	71.10	13.31
Naïve bayes	71.15	13.45	69.74	9.21	70.57	11.69	69.25	7.75

Results are given in Table 6.13. The extended features offered a performance improvement from 20.18 % error reduction to 24.26 % error reduction for the best-performing classifier (maxent). Performance with just the Reichenbach E/R determining feature are also included in the table. The feature is not as useful on its own as it is with the three other dependency-graph derived features.

The absolute increase in labelling accuracy in this subset of TLINKs is approximately 1.4 %; a modest gain, corresponding to an error reduction of. As with investigation into exploiting permanence of the reference point, problems lie in correctly identifying which of the links the features can be applied to.

6.5.3 Summary

Reichenbach's framework for tense and aspect is intuitive, and of moderate utility in typing temporal relations based on the advanced interpretation proposed above. This interpretation has already been shown to be of use when constraining TimeML interval relation types. The big question that remains is about temporal context, which has been only approximated throughout.

The framework suggests helpful constraint in cases where verbs and timexes are in the same context, already helping in automatic relation typing. However, automatic identification of where the framework applies (e.g. temporal contexts) is difficult; this is information not provided in TimeML and not trivially extractable from natural language text. An extended examination of the problems is given in [34].

As the framework is capable of capturing things that TimeML cannot and its utility can be demonstrated in controlled circumstances, it is worth investigating an extension to TimeML to improve on the standard's expressiveness by integrating ideas from Reichenbach.

6.6 Annotating Reichenbach's Framework

Existing temporal annotation schemata are not rich enough to represent all the information in Reichenbach's framework. Critically, although the framework is of use in relation typing, as demonstrated both in this book and also in recent prominent research [35], it cannot be *reliably* applied (and certainly not optimally applied) without knowledge of temporal context. In order to understand temporal context, and move towards using Reichenbach's framework effectively in temporal relation typing, this section details an annotation schema for the framework. Hopefully, given an annotation scheme, it may be possible to annotate text for temporal context and Reichenbachian tense linkages. Having annotations of temporal context enables an investigation into automatically assignment of temporal context, either by plainly revealing the rules that govern where and how contexts start and end, or by providing training data for machine learning approaches.

The new schema proposed for annotating this information is RTMML (Reichenbach Tense Model Markup Language). Following the description of the schema, we introduce a new language resource – a corpus annotated with RTMML. Finally, we demonstrate how it may be integrated with TimeML.

The annotation schema RTMML is intended to describe the verbal event structure detailed in [2], 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 that standard.

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

6.6.1 Motivation for Annotating the Framework's Points

Critical to knowing how to apply Reichenbach's framework is the issue of temporal context (Sect. 6.3.4). TimeML does not provide an annotation for this phenomenon, and so one must be introduced if we are to develop data to help understand temporal context.

Further, Reichenbach's framework also distinguishes some tenses that are ambiguous in TimeML. Given the 24 permutations for S, E, R and their relations (taken from $<$, $>$, $=$), there are 13 distinct forms, which can be further divided into tenses as below:

- Six arrangements where both relations are $=$ can be boiled down to one, through transitivity of the equality operator. (24 – 5 = 19)
- For the twelve arrangements where one relation is $=$, we halve the number of relations that we have, as the ordering of the pair of points connected by $=$ is irrelevant; for example, $S < E = R$ and $S < R = E$ are equivalent. (19 – 6 = 13)
- All arrangements where both relations are $<$ are unique and semantically distinct. (13 – 0 = 13 tenses)

TimeML's `aspect` attribute will inform us if the reference time is after the event time; that is, if the event is "complete" (to gloss over linguistic nuances detailed by [36]) before the time of reference point. This distinguishes two classes; TimeML `aspect:PERFECTIVE` corresponds to $E < R$, and `aspect:NONE` corresponds to $E \nless R$ (that is, a conflation of $E = R$ and $R < E$).

Also, TimeML does not address the issue of annotating Reichenbach's tense framework with the goal of understanding reference time or creating resources that enable detailed examination of the links between verbal events in discourse. Although other promising solutions are starting to emerge for detailed annotated of tense internals [37], it is not yet possible to describe or build relations to reference points at all in TimeML.

6.6.2 Proposed Solution

Here we discuss what should be annotated in order to capture the information described by Reichenbach's framework, and put forward an annotation schema. Some of this section's material overlaps with [38].

6.6.2.1 Requirements

A schema should allow description of the relations between the three abstract points, speech, reference and event. It must also be capable of expressing relations between different verbs' three points. Finally, it should permit events to be linked with times.

It is preferable to have a schema that follows set frameworks for linguistic annotation, hence supporting interoperability. Hopefully, this can also provide some basic structure for referencing strings within a document and an overall annotation scheme (e.g. XML).

6.6.2.2 Annotation Schema

The annotation language we propose is called RTMML, for Reichenbach Tense Model Markup Language. It includes definitions for document structure and metadata, for verb annotation, for time-referring expression annotation, and for temporal between a verb's three time points.

RTMML documents use standoff annotation. This keeps the text uncluttered, in the spirit of *ISO LAF*[4] and *ISO SemAF-Time*.[5] Annotations reference tokens by their position in the source. Token indices begin from zero. We explicitly state the segmentation plan with the `<seg>` element, as described in [39] and *ISO DIS 24614-1 WordSeg-1*.

The general speech time of a document is defined in the `<doc>` element, which has one optional attribute, `@time` (the `@` indicating that `time` is an attribute name). This is either the string `now` or a normalised value, formatted according to TIMEX3 [40] or TIDES [41].

Each `<verb>` element describes a tensed verb group – that is, a sequence of main and auxiliary verbs that comprise a single verb event. The `@target` attribute describes the verb or group's extents, using segment 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 Reichenbachian tense structure of a verb group is described using the attributes `@view` (with values *simple*, *anterior* or *posterior*) and `@tense` (*past*, *present* or *future*).

The `<verb>` element has optional attributes for directly linking a verb's speech, event or reference time to a time point specified elsewhere in the annotation. These are `@s`, `@e` and `@r` respectively. 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 well as relating to other verbs, one can reference document creation time with a value of `doc` or a temporal expression with its id (for example, `t1`).

[4]ISO 24612:2012 Language resource management – Linguistic annotation framework (LAF).

[5]ISO 24617-1:2012.

Each tensed verb has exactly one S, E and R. As these points do not hold specific values or have a position on an absolute scale, we do not attempt to directly annotate them or assign scalar values to them, instead annotating the type of relation that holds between them. For simplicity, the schema does not split E into incipitive and concluding points (these may still be expressed using TimeML if the two schemas are used in parallel).

One might think that the relations should be expressed in XML links; however this requires reifying time points. The important information is in the relations between Reichenbachian time points, with the actual temporal location of each point often never known. For this reason, the markup focuses on the relations between the Reichenbachian points for each `<verb>`, instead of attempting to assign any kind of value to individual points.

To capture these internal relations for a single verb, we use the attributes `@se`, `@er` and `@sr`. These attributes take a value that is a disjunction of $<$, $=$ and $>$ (though $<$ and $>$ are mutually exclusive). For example, `se=">"` expresses that speech time is after (succeeds) event time.

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 way as the `<doc>` element's `@time` attribute.

6.6.3 Special RTMLINKs

The `<rtmlink>` element is used to connect the speech, reference or event times between given groups of verbs. This is used, for example, for defining a temporal context between verbs that have the same reference time, or annotating positional use of the reference point where a given timex described the reference point of a particular verb event.

To simplify the annotation task, RTMML permits an alternative annotation with the `<rtmlink>` element. The `<rtmlink>` annotation can be used to describe verbs affected by permanence of the reference point (e.g. to reify temporal contexts), positional use of the reference point and positional use of the speech point. 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 6.14.

Table 6.14 RTMML relation types

Relation name	Description	Interpretation
POSITIONS	Reference point is set by a timex	$T_a = R_b$
SAME_TIMEFRAME	Verbs in the same temporal context	$R_a = R_b[, R_c, \ldots R_x]$
REPORTS	Reported speech or events	$E_a = S_b$

When more than two entities are listed as `rtmlink` 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.

6.6.4 Example RTMML

This section includes worked examples of sentences and their RTMML annotations.

In Example 32, we define a time *Yesterday* as `t1` and a verbal event *ate* as `v1`.

Example 32
```
<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>
```

The tense of `v1` is placed 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, that is, where a time-referring expression determines reference time. This is annotated in `v1` when we say `r="t1"` to verbosely capture a use of the reference point. Further, as the default S/E/R structure of a Reichenbachian simple past tensed verb is non-ambiguous, the attributes signifying relations between time points may be omitted. To simplify the RTMML in Example 32, we could replace the `<verb>` element with that in Example 33:

Example 33
```
<verb xml:id="v1"target="#token3"
      view="simple"tense="past"
      s="doc"/>
<rtmlink xml:id="l1"type="POSITIONS">
  <link source="#t1"/>
```

```
  <link target="#v1"/>
</rtmlink>
```

Longer examples can be found in the appendices, including an excerpt of David Copperfield in Example 34 and Fig. B.1.

6.6.4.1 Comments on Annotation

As can be seen in Table 6.2, 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 from a specific example how to resolve such an ambiguity. In other cases, 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 possible relation types; continuing the simple future example, we could say "$S < R$ or $S = R$" with `sr="<="`.

Some parts of the annotation task present difficulties. During a trial annotation, while annotators could determine the scoping exercise that is temporal context annotation without too much difficulty, directly mapping a verb group to a single Reichenbachian tense schema was hard, and at best tiring. Decomposing this task into pairwise judgements between S, E and R made annotation easier, though when one could often not see all the information required in order to make the correct judgement; as a result, many pairwise annotations were changed after annotators considered distinct but related pairs. Posing the annotation task as one of temporal constraint, using more concrete ideas (e.g. "From the text, does this event of *John running* obtain at 9p.m.?" instead of "Is T_9 during E_7?") may reduce annotator fatigue and error. RTMML does not address intentionality, leaving this to annotators and, where expressable, TimeML (which includes the I_ACTION and I_STATE event classes for this purpose).

RTMML annotation is also independent of language. As long as a segmentation scheme (e.g. WordSeg-1) is agreed, the model can be applied and an annotation created.

6.6.4.2 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 `<EVENT>` elements, and `<rtmlink>`s will refer to event or event instance IDs.

6.7 Chapter Summary

Previous findings suggested that tense shifts played a significant part in temporal relation typing, especially of difficult links. To this end, in this chapter, we introduced Reichenbach's framework for tense and aspect. The chapter introduced novel additions to the framework, and proposed two interpretations of it (one minimal, one advanced) in the context of TimeML. The advanced interpretation was used to perform the first validation of Reichenbach's framework against gold-standard temporally annotated resources, and provided empirical support for Reichenbach's 65-year-old theoretical framework. While showing support for the framework, the validation also uncovered important issues regarding how to choose which events or times could be linked, which is described in this book as "temporal context".

Given the framework, a method of interpreting it and a demonstration of its validity, this chapter also investigated how to leverage the framework in the overall problem of the relation typing task. Various approaches to using Reichenbach's framework in machine learning approaches to temporal relation typing were described. This allowed experimentation with different approximations of temporal context, and showed that the framework can be leveraged for real temporal relation typing gains.

These empirical results supported a further investigation into temporal context, which is begun with the introduction in this chapter of an annotation schema for Reichenbach's framework, that permits not only delineation of temporal context bounds but also annotation of reference time, as well as speech and event times in a corpus.

References

1. Lapata, M., Lascarides, A.: Learning sentence-internal temporal relations. J. Artif. Intell. Res. **27**(1), 85–117 (2006)
2. Reichenbach, H.: The tenses of verbs. In: Elements of Symbolic Logic. Dover Publications (1947)
3. Harris, R., Brewer, W.: Deixis in memory for verb tense. J. Verbal Learn. Verbal Behav. **12**(5), 590–597 (1973)
4. Hepple, M., Setzer, A., Gaizauskas, R.: USFD: preliminary exploration of features and classifiers for the TempEval-2007 tasks. In: Proceedings of the 4th International Workshop on Semantic Evaluations, SemEval 2007, pp. 438–441. Association for Computational Linguistics (2007)
5. Pratchett, T.: The Light Fantastic. Colin Smythe, Gerards Cross (1986)
6. Matlock, T., Ramscar, M., Boroditsky, L.: On the experiential link between spatial and temporal language. Cogn. Sci. **29**(4), 655–664 (2005)
7. Huggett, N.: Zeno's paradoxes. In: Zalta, E.N. (ed.) The Stanford Encyclopedia of Philosophy (Winter), 2010 edn. CSLI (2010)
8. Eddington, A.: The Nature of the Physical World. Macmillen, Cambridge (1928)
9. Stocker, K.: The time machine in our mind. Cogn. Sci. **36**, 385–420 (2012)
10. McTaggart, J.: The unreality of time. Mind **17**(4), 457 (1908)
11. Lyons, J.: Semantics, vol. 2. Cambridge University Press, Cambridge (1977)

12. Michaelis, L.: Time and tense. The Handbook of English Linguistics, pp. 220–243. Blackwell, Oxford (2006)
13. Jaszczolt, K.M.: Representing Time: An Essay on Temporality as Modality. Oxford University Press, Oxford (2009)
14. Fillmore, C.: Lectures on Deixis. CSLI Publications Stanford, California (1971)
15. Klein, W.: Time in Language. Germanic Linguistics. Routledge, London (1994)
16. Mani, I., Pustejovsky, J., Gaizauskas, R.: The Language of Time: A Reader. Oxford University Press, Oxford (2005)
17. Song, F., Cohen, R.: The interpretation of temporal relations in narrative. In: Proceedings of the 7th National Conference of AAAI (1988)
18. Ahn, D., Adafre, S., Rijke, M.: Towards task-based temporal extraction and recognition. In: Dagstuhl Seminar Proceedings, vol. 5151 (2005)
19. Hornstein, N.: As Time goes by: Tense and Universal Grammar. MIT Press, Cambridge (1990)
20. Dowty, D.: The effects of aspectual class on the temporal structure of discourse: semantics or pragmatics? Linguist. Philos. **9**(1), 37–61 (1986)
21. Miller, T.A., Bethard, S., Dligach, D., Pradhan, S., Lin, C., Savova, G.K.: Discovering narrative containers in clinical text. ACL **2013**, 18 (2013)
22. By, T.: Tears in the rain. Ph.D. thesis, University of Sheffield (2002)
23. Paslawska, A., van Stechow, A.: Perfect readings in russian. Perfect Explorations **2**, 307 (2003)
24. Hinrichs, E.: Temporal anaphora in discourses of english. Linguist. Philos. **9**(1), 63–82 (1986)
25. Giorgi, A., Pianesi, F.: Tense and Aspect: From Semantics to Morphosyntax. Oxford University Press, USA (1997)
26. Hristova, D.: The neoreichenbachian model of tense syntax and the rusian active participles. Harv. Ukr. Stud. **28**(1/4), 155–164 (2006)
27. Prior, A.: Past, Present and Future. Clarendon, Oxford (1967)
28. Tanaka, K.: On reichenbach's approach to tense. Tsukuba Engl. Stud. **9**, 61–75 (1990)
29. Derczynski, L., Gaizauskas, R.: Empirical validation of reichenbach's tense framework. In: Proceedings of the 10th Conference on Computational Semantics, pp. 71–82. Association for Computational Linguistics (2013)
30. Kowalski, R., Sergot, M.: A logic-based calculus of events. In: Foundations of Knowledge Base Management, pp. 23–55. Springer (1989)
31. Costa, F., Branco, A.: Aspectual type and temporal relation classification. In: Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, pp. 266–275 (2012)
32. Kiss, T., Strunk, J.: Unsupervised multilingual sentence boundary detection. Comput. Linguist. **32**(4), 485–525 (2006)
33. De Marneffe, M., MacCartney, B., Manning, C.: Generating typed dependency parses from phrase structure parses. In: Proceedings of the International Conference on Language Resources and Evaluation (2006)
34. Derczynski, L., Gaizauskas, R.: Temporal relation classification using a model of tense and aspect. In: Proceedings of the Conference on Recent Advances in Natural Language Processing. Association for Computational Linguistics (2015)
35. Chambers, N., Cassidy, T., McDowell, B., Bethard, S.: Dense event ordering with a multi-pass architecture. Trans. Assoc. Comput. Linguist. **2**, 273–284 (2014)
36. Vendler, Z.: Verbs and times. Philos. Rev. **66**(2), 143–160 (1957)
37. Gast, V., Bierkandt, L., Rzymski, C.: Creating and retrieving tense and aspect annotations with GraphAnno, a lightweight tool for multi-level annotation. In: Proceedings 11th Joint ACL-ISO Workshop on Interoperable Semantic Annotation (ISA-11), pp. 23–28 (2015)
38. Derczynski, L., Gaizauskas, R.: An Annotation Scheme for Reichenbach's Verbal Tense Structure. In: Workshop on Interoperable Semantic Annotation, pp. 10–17 (2011)
39. Lee, K., Romary, L.: Towards Interoperability of ISO Standards for Language Resource Management. In: International Conference on Global Interoperability for Language Resources (2010)

40. Boguraev, B., Ando, R.: TimeML-compliant text analysis for temporal reasoning. In: Proceedings of International Joint Conference on Artificial Intelligence (IJCAI) (2005)
41. Ferro, L., Gerber, L., Mani, I., Sundheim, B., Wilson, G.: Tides 2005 standard for the annotation of temporal expressions. Technical report 03–1046, The MITRE Corporation (2005)