CHAPTER

22 Time and Temporal Reasoning

Time will explain.
Jane Austen, *Persuasion*

When we discussed events back in Chapter 19 and Chapter 21, we temporally delayed discussing **time**. But as they say, time waits for no one, and there's no time like the present! Let's take the time now to talk about representing and extracting time. Events are situated in time, occurring at a particular date or time, and events can be related temporally, happening before or after or simultaneously with each other. Consider what we would need to understand such temporal issues in this shortened version of the news story we saw in Chapter 21:

> United Airlines said Friday it has increased fares by $6 per round trip on flights to some cities also served by lower-cost carriers. American Airlines, a unit of AMR Corp., immediately matched the move, spokesman Tim Wagner said. United, a unit of UAL Corp., said the increase took effect Thursday.

We'll need to recognize temporal expressions like days of the week (*Friday* and *Thursday*) or *two days from now* and times such as *3:30 P.M.*, and **normalize** them onto specific calendar dates or times. We'll need to link *Friday* to the time of United's announcement, *Thursday* to the previous day's fare increase, and we'll need to produce a timeline in which United's announcement follows the fare increase and American's announcement follows both of those events. And we'll need to base all this on theories of how time and aspect work, how to represent how events are located in time and their temporal relation to each other. And we'll need to be able to practically extract all these things from text. Not all of these tasks are solvable in current systems, but we'll explore what is currently doable and what questions remain open.

22.1 Representing Time

temporal logic

Let's begin by introducing the basics of **temporal logic** and how human languages convey temporal information. The most straightforward theory of time holds that it flows inexorably forward and that events are associated with either points or intervals in time, as on a timeline. We can order distinct events by situating them on the timeline; one event *precedes* another if the flow of time leads from the first event to the second. Accompanying these notions in most theories is the idea of the current moment in time. Combining this notion with the idea of a temporal ordering relationship yields the familiar notions of past, present, and future.

Many schemes can represent this kind of temporal information. The one presented here is a fairly simple one that stays within the FOL framework of reified events that we pursued in Chapter 19. Consider the following examples:

(22.1) I arrived in New York.

(22.2) I am arriving in New York.

(22.3) I will arrive in New York.

These sentences all refer to the same kind of event and differ solely in the verb tense. In the Davidsonian scheme for representing events (Chapter 19), all three would share the following representation, which lacks any temporal information:

$$\exists e Arriving(e) \wedge Arriver(e, Speaker) \wedge Destination(e, NewYork) \qquad (22.4)$$

The temporal information provided by the tense of the verbs can be exploited by predicating additional information about the event variable e. Various kinds of temporal logics can be used to talk about temporal ordering relationship. One of the most commonly used in computational modeling is the **interval algebra** of Allen (1984). Allen models all events and time expressions as intervals there is no representation for points (although intervals can be very short). In order to deal with intervals without points, he identifies 13 primitive relations that can hold between these temporal intervals. Fig. 22.1 shows these 13 **Allen relations**.

interval algebra

Allen relations

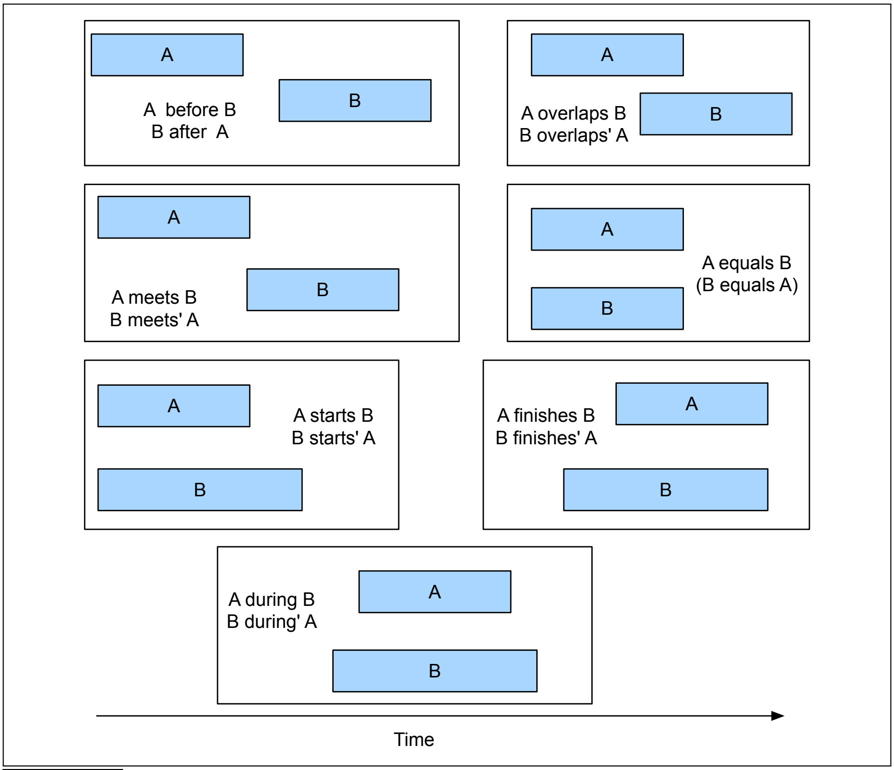

Figure 22.1 The 13 temporal relations from Allen (1984).

To include the interval algebra in our model, we add a temporal variable that represents the interval corresponding to the event, another (very small) interval corresponding to the current time *Now*, and temporal predicates relating the event to the current time as indicated by the tense of the verb. This interval algebra approach yields the following representations for our *arriving* examples:

$$
\begin{aligned}
\exists e,i\, Arriving(e) \;&\wedge\; Arriver(e,Speaker) \wedge Destination(e,NewYork)\\
&\wedge\; IntervalOf(e,i) \wedge Before(i,Now)\\
\exists e,i\, Arriving(e) \;&\wedge\; Arriver(e,Speaker) \wedge Destination(e,NewYork)\\
&\wedge\; IntervalOf(e,i) \wedge During(i,Now)\\
\exists e,i\, Arriving(e) \;&\wedge\; Arriver(e,Speaker) \wedge Destination(e,NewYork)\\
&\wedge\; IntervalOf(e,i) \wedge After(i,Now)
\end{aligned}
$$

In addition to the variable *i* that stands for the interval of time associated with the event, we see the two-place predicate *Before* that represents the notion that the first interval argument precedes the second in time and the constant *Now* that refers to the interval corresponding to the current time. For past events, the interval must end before the current time. Similarly, for future events the current time must precede the end of the event. For events happening in the present, the current time is contained within the event interval, using the predicate *During*. We'll see later in this chapter how these predicates can be used to detect and link the temporal relations between events in a text to give us a complete timeline.

22.1.1 Reichenbach's reference point

The relation between simple verb tenses and points in time is by no means straightforward. The present tense can be used to refer to a future event, as in this example:

(22.5) Ok, we fly from San Francisco to Boston at 10.

Or consider the following examples:

(22.6) Flight 1902 arrived late.

(22.7) Flight 1902 had arrived late.

Although both refer to events in the past, representing them in the same way seems wrong. The second example seems to have another unnamed event lurking in the background (e.g., Flight 1902 had already arrived late *when* something else happened).

To account for this phenomena, Reichenbach (1947) introduced the notion of a **reference point**. In our simple temporal scheme, the current moment in time is equated with the time of the utterance and is used as a reference point for when the event occurred (before, at, or after). In Reichenbach's approach, the notion of the reference point is separated from the utterance time and the event time. The following examples illustrate the basics of this approach:

reference point

(22.8) When Mary's flight departed, I ate lunch.

(22.9) When Mary's flight departed, I had eaten lunch.

In both of these examples, the eating event has happened in the past, that is, prior to the utterance. However, the verb tense in the first example indicates that the eating event began when the flight departed, while the second example indicates that the eating was accomplished prior to the flight's departure. Therefore, in Reichenbach's terms the *departure* event specifies the reference point. These facts can be accommodated by additional constraints relating the *eating* and *departure* events. In the first example, the reference point precedes the *eating* event, and in the second example, the eating precedes the reference point. Figure 22.2 illustrates Reichenbach's approach with the primary English tenses. Exercise 22.2 asks you to represent these examples in FOL.

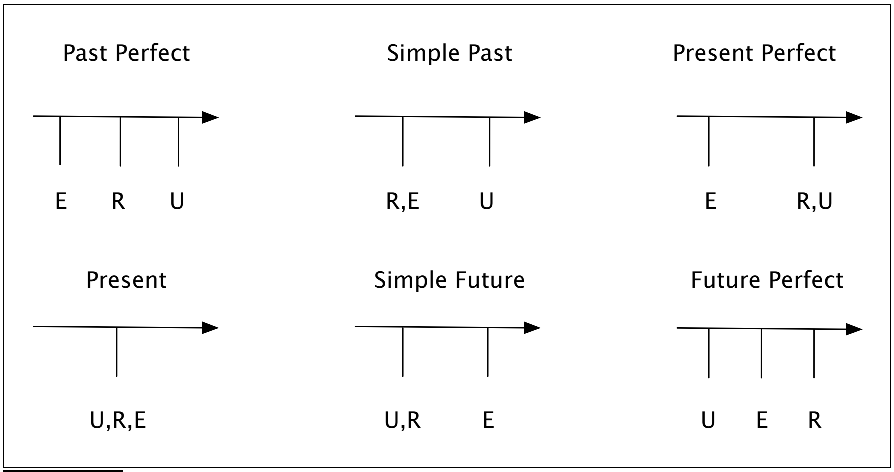

Figure 22.2 Reichenbach's approach applied to various English tenses. In these diagrams, time flows from left to right, **E** denotes the time of the event, **R** denotes the reference time, and **U** denotes the time of the utterance.

Languages have many other ways to convey temporal information besides tense. Most useful for our purposes will be temporal expressions like *in the morning* or *6:45* or *afterwards*.

(22.10) I'd like to go at 6:45 in the morning.

(22.11) Somewhere around noon, please.

(22.12) I want to take the train back afterwards.

Incidentally, temporal expressions display a fascinating metaphorical conceptual organization. Temporal expressions in English are frequently expressed in spatial terms, as is illustrated by the various uses of *at*, *in*, *somewhere*, and *near* in these examples (Lakoff and Johnson 1980, Jackendoff 1983). Metaphorical organizations such as these, in which one domain is systematically expressed in terms of another, are very common in languages of the world.

22.2 Representing Aspect

aspect

A related notion to time is **aspect**, which is what we call the way events can be categorized by their internal temporal structure or temporal contour. By this we mean questions like whether events are ongoing or have ended, or whether they are conceptualized as happening at a point in time or over some interval. Such notions of temporal contour have been used to divide event expressions into classes since Aristotle, although the set of four classes we'll introduce here is due to Vendler (1967) (you may also see the German term **aktionsart** used to refer to these classes).

aktionsart **events** **states** **stative**

The most basic aspectual distinction is between **events** (which involve change) and **states** (which do not involve change). **Stative expressions** represent the notion of an event participant being in a **state**, or having a particular property, at a given point in time. Stative expressions capture aspects of the world at a single point in time, and conceptualize the participant as unchanging and continuous. Consider the following ATIS examples.

(22.13) I like express trains.

(22.14) I need the cheapest fare.

(22.15) I want to go first class.

In examples like these, the event participant denoted by the subject can be seen as experiencing something at a specific point in time, and don't involve any kind of internal change over time (the liking or needing is conceptualized as continuous and unchanging).

Non-states (which we'll refer to as **events**) are divided into subclasses; we'll introduce three here. **Activity expressions** describe events undertaken by a participant that occur over a span of time (rather than being conceptualized as a single point in time like stative expressions), and have no particular end point. Of course in practice all things end, but the meaning of the expression doesn't represent this fact. Consider the following examples:

activity

(22.16) She drove a Mazda.

(22.17) I live in Brooklyn.

These examples both specify that the subject is engaged in, or has engaged in, the activity specified by the verb for some period of time, but doesn't specify when the driving or living might have stopped.

Two more classes of expressions, **achievement** expressions and **accomplishment** expressions, describe events that take place over time, but also conceptualize the event as having a particular kind of endpoint or goal. The Greek word *telos* means 'end' or 'goal' and so the events described by these kinds of expressions are often called **telic** events.

telic

accomplishment expressions

Accomplishment expressions describe events that have a natural end point and result in a particular state. Consider the following examples:

(22.18) He booked me a reservation.

(22.19) The 7:00 train got me to New York City.

In these examples, an event is seen as occurring over some period of time that ends when the intended state is accomplished (i.e., the state of me having a reservation, or me being in New York City).

achievement expressions

The final aspectual class, **achievement expressions**, is only subtly different than accomplishments. Consider the following:

(22.20) She found her gate.

(22.21) I reached New York.

Like accomplishment expressions, achievement expressions result in a state. But unlike accomplishments, achievement events are 'punctual': they are thought of as happening in an instant and the verb doesn't conceptualize the process or activity leading up the state. Thus the events in these examples may in fact have been preceded by extended *searching* or *traveling* events, but the verb doesn't conceptualize these preceding processes, but rather conceptualizes the events corresponding to *finding* and *reaching* as points, not intervals.

In summary, a standard way of categorizing event expressions by their temporal contours is via these four general classes:

Stative: I know my departure gate.

Activity: John is flying.

Accomplishment: Sally booked her flight.

Achievement: She found her gate.

Before moving on, note that event expressions can easily be shifted from one class to another. Consider the following examples:

(22.22) I flew.

(22.23) I flew to New York.

The first example is a simple activity; it has no natural end point. The second example is clearly an accomplishment event since it has an end point, and results in a particular state. Clearly, the classification of an event is not solely governed by the verb, but by the semantics of the entire expression in context.

22.3 Temporally Annotated Datasets: TimeBank

TimeBank

The **TimeBank** corpus consists of American English text annotated with temporal information (Pustejovsky et al., 2003). The annotations use TimeML (Saurí et al., 2006), a markup language for time based on Allen's interval algebra discussed above (Allen, 1984). There are three types of TimeML objects: an EVENT represent events and states, a TIME represents time expressions like dates, and a LINK represents various relationships between events and times (event-event, event-time, and time-time). The links include temporal links (TLINK) for the 13 Allen relations, aspectual links (ALINK) for aspectual relationships between events and subevents, and SLINKS which mark factuality.

Consider the following sample sentence and its corresponding markup shown in Fig. 22.3, selected from one of the TimeBank documents.

(22.24) Delta Air Lines earnings soared 33% to a record in the fiscal first quarter, bucking the industry trend toward declining profits.

```
<TIMEX3 tid="t57" type="DATE" value="1989-10-26"  functionInDocument="CREATION_TIME">
10/26/89  </TIMEX3>

Delta Air Lines earnings <EVENT eid="e1" class="OCCURRENCE"> soared </EVENT> 33% to a
record in  <TIMEX3 tid="t58" type="DATE" value="1989-Q1" anchorTimeID="t57"> the
fiscal first quarter </TIMEX3>, <EVENT eid="e3"  class="OCCURRENCE">bucking</EVENT>
the industry trend toward <EVENT eid="e4" class="OCCURRENCE">declining</EVENT>
profits.
```

Figure 22.3 Example from the TimeBank corpus.

This text has three events and two temporal expressions (including the creation time of the article, which serves as the document time), and four temporal links that capture the using the Allen relations:

- Soaring$_{e1}$ is **included** in the fiscal first quarter$_{t58}$
- Soaring$_{e1}$ is **before** 1989-10-26$_{t57}$
- Soaring$_{e1}$ is **simultaneous** with the bucking$_{e3}$
- Declining$_{e4}$ **includes** soaring$_{e1}$

We can also visualize the links as a graph. The TimeBank snippet in Eq. 22.25 would be represented with a graph like Fig. 22.4.

(22.25) **[DCT:11/02/891]**$_1$: Pacific First Financial Corp. **said**$_2$ shareholders **approved**$_3$ its **acquisition**$_4$ by Royal Trustco Ltd. of Toronto for $27 a share, or $212 million. The thrift holding company **said**$_5$ it **expects**$_6$ to **obtain**$_7$ regulatory **approval**$_8$ and **complete**$_9$ the **transaction**$_{10}$ by **year-end**$_{11}$.

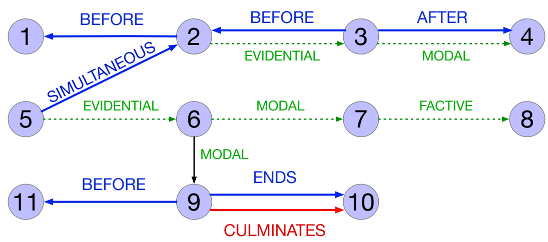

Figure 22.4 A graph of the text in Eq. 22.25, adapted from (Ocal et al., 2022). TLINKS are shown in blue, ALINKS in red, and SLINKS in green.

22.4 Automatic Temporal Analysis

Here we introduce the three common steps used in analyzing time in text:

1. Extracting **temporal expressions**
2. **Normalizing** these expressions, by converting them to a standard format.
3. **Linking** events to times and extracting time graphs and timelines

22.4.1 Extracting Temporal Expressions

Temporal expressions are phrases that refer to absolute points in time, relative times, durations, and sets of these. **Absolute** temporal expressions are those that can be mapped directly to calendar dates, times of day, or both. **Relative** temporal expressions map to particular times through some other reference point (as in *a week from last Tuesday*). Finally, **durations** denote spans of time at varying levels of granularity (seconds, minutes, days, weeks, centuries, etc.). Figure 22.5 lists some sample temporal expressions in each of these categories.

absolute

relative

duration

Absolute	Relative	Durations
April 24, 1916	yesterday	four hours
The summer of '77	next semester	three weeks
10:15 AM	two weeks from yesterday	six days
The 3rd quarter of 2006	last quarter	the last three quarters

Figure 22.5 Examples of absolute, relational and durational temporal expressions.

Temporal expressions are grammatical constructions that often have temporal **lexical triggers** as their heads, making them easy to find. Lexical triggers might be nouns, proper nouns, adjectives, and adverbs; full temporal expressions consist of their phrasal projections: noun phrases, adjective phrases, and adverbial phrases (Figure 22.6).

lexical triggers

The task is to detect temporal expressions in running text, like this examples, shown with TIMEX3 tags (Pustejovsky et al. 2005, Ferro et al. 2005).

Category	Examples
Noun	*morning, noon, night, winter, dusk, dawn*
Proper Noun	*January, Monday, Ides, Easter, Rosh Hashana, Ramadan, Tet*
Adjective	*recent, past, annual, former*
Adverb	*hourly, daily, monthly, yearly*

Figure 22.6 Examples of temporal lexical triggers.

> A fare increase initiated <TIMEX3>last week</TIMEX3> by UAL Corp's United Airlines was matched by competitors over <TIMEX3>the weekend</TIMEX3>, marking the second successful fare increase in <TIMEX3>two weeks</TIMEX3>.

Rule-based approaches use cascades of regular expressions to recognize larger and larger chunks from previous stages, based on patterns containing parts of speech, trigger words (e.g., *February*) or classes (e.g., *MONTH*) (Chang and Manning, 2012; Strötgen and Gertz, 2013; Chambers, 2013). Here's a rule from SUTime (Chang and Manning, 2012) for detecting expressions like *3 years old*:

```
/(\d+)[-\s]($TEUnits)(s)?([-\s]old)?/
```

Sequence-labeling approaches use the standard IOB scheme, marking words that are either (I)nside, (O)utside or at the (B)eginning of a temporal expression:

A fare increase initiated last week by UAL Corp's...
O O O O B I O O O

A statistical sequence labeler is trained, using either embeddings or a fine-tuned encoder, or classic features extracted from the token and context including words, lexical triggers, and POS.

Temporal expression recognizers are evaluated with the usual recall, precision, and *F*-measures. A major difficulty for all of these very lexicalized approaches is avoiding expressions that trigger false positives:

(22.26) *1984* tells the story of Winston Smith...

(22.27) ...U2's classic *Sunday* Bloody *Sunday*

22.4.2 Temporal Normalization

temporal normalization

Temporal normalization is the task of mapping a temporal expression to a point in time or to a duration. Points in time correspond to calendar dates, to times of day, or both. Durations primarily consist of lengths of time. Normalized times are represented via the ISO 8601 standard for encoding temporal values (ISO8601, 2004). Fig. 22.7 reproduces our earlier example with these value attributes.

```
<TIMEX3 id="t1'' type="DATE" value="2007-07-02" functionInDocument="CREATION_TIME">
   July 2, 2007 </TIMEX3> A fare increase initiated <TIMEX3 id="t2" type="DATE"
   value="2007-W26" anchorTimeID="t1">last week</TIMEX3> by United Airlines was
   matched by competitors over <TIMEX3 id="t3" type="DURATION" value="P1WE"
   anchorTimeID="t1"> the weekend </TIMEX3>, marking the second successful fare
   increase in <TIMEX3 id="t4" type="DURATION" value="P2W" anchorTimeID="t1"> two
   weeks </TIMEX3>.
```

Figure 22.7 TimeML markup including normalized values for temporal expressions.

The dateline, or document date, for this text was *July 2, 2007*. The ISO representation for this kind of expression is YYYY-MM-DD, or in this case, 2007-07-02. The encodings for the temporal expressions in our sample text all follow from this date, and are shown here as values for the VALUE attribute.

The first temporal expression in the text proper refers to a particular week of the year. In the ISO standard, weeks are numbered from 01 to 53, with the first week of the year being the one that has the first Thursday of the year. These weeks are represented with the template YYYY-Wnn. The ISO week for our document date is week 27; thus the value for *last week* is represented as "2007-W26".

The next temporal expression is *the weekend*. ISO weeks begin on Monday; thus, weekends occur at the end of a week and are fully contained within a single week. Weekends are treated as durations, so the value of the VALUE attribute has to be a length. Durations are represented according to the pattern P*nx*, where *n* is an integer denoting the length and *x* represents the unit, as in P3Y for *three years* or P2D for *two days*. In this example, one weekend is captured as P1WE. In this case, there is also sufficient information to anchor this particular weekend as part of a particular week. Such information is encoded in the ANCHORTIMEID attribute. Finally, the phrase *two weeks* also denotes a duration captured as P2W. Figure 22.8 give some more examples, but there is a lot more to the various temporal annotation standards; consult ISO8601 (2004), Ferro et al. (2005), and Pustejovsky et al. (2005) for more details.

Unit	Pattern	Sample Value
Fully specified dates	YYYY-MM-DD	1991-09-28
Weeks	YYYY-Wnn	2007-W27
Weekends	PnWE	P1WE
24-hour clock times	HH:MM:SS	11:13:45
Dates and times	YYYY-MM-DDTHH:MM:SS	1991-09-28T11:00:00
Financial quarters	Qn	1999-Q3

Figure 22.8 Sample ISO patterns for representing various times and durations.

Most current approaches to temporal normalization are rule-based (Chang and Manning 2012, Strötgen and Gertz 2013). Patterns that match temporal expressions are associated with semantic analysis procedures. For example, the pattern above for recognizing phrases like *3 years old* can be associated with the predicate *Duration* that takes two arguments, the length and the unit of time:

```
pattern: /(\d+)[-\s]($TEUnits)(s)?([-\s]old)?/
result: Duration($1, $2)
```

The task is difficult because fully qualified temporal expressions are fairly rare in real texts. Most temporal expressions in news articles are incomplete and are only implicitly anchored, often with respect to the dateline of the article, which we refer to as the document's **temporal anchor**. The values of temporal expressions such as *today*, *yesterday*, or *tomorrow* can all be computed with respect to this temporal anchor. The semantic procedure for *today* simply assigns the anchor, and the attachments for *tomorrow* and *yesterday* add a day and subtract a day from the anchor, respectively. Of course, given the cyclic nature of our representations for months, weeks, days, and times of day, our temporal arithmetic procedures must use modulo arithmetic appropriate to the time unit being used.

temporal anchor

Unfortunately, even simple expressions such as *the weekend* or *Wednesday* introduce a fair amount of complexity. In our current example, *the weekend* clearly refers to the weekend of the week that immediately precedes the document date. But this won't always be the case, as is illustrated in the following example.

(22.28) Random security checks that began yesterday at Sky Harbor will continue at least through the weekend.

In this case, the expression *the weekend* refers to the weekend of the week that the anchoring date is part of (i.e., the coming weekend). The information that signals this meaning comes from the tense of *continue*, the verb governing *the weekend*.

Relative temporal expressions are handled with temporal arithmetic similar to that used for *today* and *yesterday*. The document date indicates that our example article is ISO week 27, so the expression *last week* normalizes to the current week minus 1. To resolve ambiguous *next* and *last* expressions we consider the distance from the anchoring date to the nearest unit. *Next Friday* can refer either to the immediately next Friday or to the Friday following that, but the closer the document date is to a Friday, the more likely it is that the phrase will skip the nearest one. Such ambiguities are handled by encoding language and domain-specific heuristics into the temporal attachments.

22.4.3 Temporal Ordering of Events

The goal of temporal analysis, is to link times to events and then fit all these events into a complete timeline. This ambitious task is the subject of considerable current research but solving it with a high level of accuracy is beyond the capabilities of current systems. A somewhat simpler, but still useful, task is to impose a partial ordering on the events and temporal expressions mentioned in a text. Such an ordering can provide many of the same benefits as a true timeline. An example of such a partial ordering is the determination that the fare increase by *American Airlines* came *after* the fare increase by *United* in our sample text. Determining such an ordering can be viewed as a binary relation detection and classification task similar to those described in Chapter 21.

Even this partial ordering task assumes that in addition to the detecting and normalizing time expressions steps described above, we have already detected all the events in the text using the methods we saw in Chapter 21. Indeed, many temporal expressions are anchored to events mentioned in a text and not directly to other temporal expressions. Consider the following example:

(22.29) One week after the storm, JetBlue issued its customer bill of rights.

To determine when JetBlue issued its customer bill of rights we need to determine the time of *the storm* event, and then we need to modify that time by the temporal expression *one week after*.

Thus once the events and times have been detected, our goal next is to assert links between all the times and events: i.e. creating event-event, event-time, time-time, DCT-event, and DCT-time TimeML TLINKS. This can be done by training time relation classifiers to predict the correct TLINK between each pair of times/events, supervised by the gold labels in the TimeBank corpus with features like words/embeddings, parse paths, tense and aspect The sieve-based architecture using precision-ranked sets of classifiers, which we'll introduce in Chapter 26, is also commonly used.

Systems that perform all 4 tasks (time extraction creation and normalization, event extraction, and time/event linking) include TARSQI (Verhagen et al., 2005) CLEARTK (Bethard, 2013), CAEVO (Chambers et al., 2014), and CATENA (Mirza and Tonelli, 2016).

22.5 Summary

This chapter has introduced ways of representing, extracting, and reasoning about time. The following are some of the highlights of this chapter:

- Reasoning about time can be facilitated by detection and normalization of **temporal expressions**.
- **Events** can be ordered in time using sequence models and classifiers trained on temporally- and event-labeled data like the **TimeBank corpus**.

Bibliographical and Historical Notes

Exercises

22.1 A useful functionality in newer email and calendar applications is the ability to associate temporal expressions connected with events in email (doctor's appointments, meeting planning, party invitations, etc.) with specific calendar entries. Collect a corpus of email containing temporal expressions related to event planning. How do these expressions compare to the kinds of expressions commonly found in news text that we've been discussing in this chapter?

22.2 For the following sentences, give FOL translations that capture the temporal relationships between the events.

1. When Mary's flight departed, I ate lunch.
2. When Mary's flight departed, I had eaten lunch.

Allen, J. 1984. Towards a general theory of action and time. *Artificial Intelligence*, 23(2):123–154.

Bethard, S. 2013. ClearTK-TimeML: A minimalist approach to TempEval 2013. *SemEval-13*.

Chambers, N. 2013. NavyTime: Event and time ordering from raw text. *SemEval-13*.

Chambers, N., T. Cassidy, B. McDowell, and S. Bethard. 2014. Dense event ordering with a multi-pass architecture. *TACL*, 2:273–284.

Chang, A. X. and C. D. Manning. 2012. SUTime: A library for recognizing and normalizing time expressions. *LREC*.

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

ISO8601. 2004. Data elements and interchange formats—information interchange—representation of dates and times. Technical report, International Organization for Standards (ISO).

Jackendoff, R. 1983. *Semantics and Cognition*. MIT Press.

Lakoff, G. and M. Johnson. 1980. *Metaphors We Live By*. University of Chicago Press, Chicago, IL.

Mirza, P. and S. Tonelli. 2016. CATENA: CAusal and TEmporal relation extraction from NAtural language texts. *COLING*.

Ocal, M., A. Perez, A. Radas, and M. Finlayson. 2022. Holistic evaluation of automatic TimeML annotators. *LREC*.

Pustejovsky, J., P. Hanks, R. Saurí, A. See, R. Gaizauskas, A. Setzer, D. Radev, B. Sundheim, D. S. Day, L. Ferro, and M. Lazo. 2003. The TIMEBANK corpus. *Proceedings of Corpus Linguistics 2003 Conference*. UCREL Technical Paper number 16.

Pustejovsky, J., R. Ingria, R. Saurí, J. Castaño, J. Littman, R. Gaizauskas, A. Setzer, G. Katz, and I. Mani. 2005. *The Specification Language TimeML*, chapter 27. Oxford.

Reichenbach, H. 1947. *Elements of Symbolic Logic*. Macmillan, New York.

Saurí, R., J. Littman, B. Knippen, R. Gaizauskas, A. Setzer, and J. Pustejovsky. 2006. TimeML annotation guidelines version 1.2.1. Manuscript.

Strötgen, J. and M. Gertz. 2013. Multilingual and cross-domain temporal tagging. *Language Resources and Evaluation*, 47(2):269–298.

Vendler, Z. 1967. *Linguistics in Philosophy*. Cornell University Press.

Verhagen, M., I. Mani, R. Sauri, R. Knippen, S. B. Jang, J. Littman, A. Rumshisky, J. Phillips, and J. Pustejovsky. 2005. Automating temporal annotation with TARSQI. *ACL*.