Chapter 2
Events and Times

> Hvor er jeg? Hvad vil det sige:Verden? Hvad betyder dette Ord?
> Hvo har narret mig ind i det Hele, og lader mig nu staae der?
> *Where am I? What does it mean to say: the world? What is the meaning of that word? Who tricked me into this whole thing and leaves me standing here?*
>
> *Repetition*
> SØREN KIERKEGAARD

2.1 Introduction

Time is a critical part of language. Without the ability to express it, we cannot plan, tell stories or discuss change. Almost all empirical assertions are transient and have temporal bounds; because of this the capability to describe the future, the past and the present is critical to accurate information transfer through language.

If we are to have a computer reason about times and events, we need to know about time in language. Time in language can be broken down into three primitives: times, events and temporal relations [1]. Viewing the temporal structure of a discourse as a graph, the times and events are the nodes and the relations the arcs. In this chapter, we introduce the nodes – events and times.

Some theories and models of language include or focus on temporality. While some linguistic theories related to time require a human-level understanding of text, others use very finite terms which operate using features of language that we can already automatically identify with a high degree of confidence. Based on these linguistic theories, we can describe certain structures in text as well as their behaviour. We may leverage this to better understand and process temporal information in discourse.

Finally, given this understanding, it is possible to build systems for some automatic temporal processing. There are approaches to detecting times and events, to determining event durations [2, 3] and to typing the relation between two events [4–6].

L.R.A. Derczynski, *Automatically Ordering Events and Times in Text*,
Studies in Computational Intelligence 677, DOI 10.1007/978-3-319-47241-6_2

This chapter presents background material relevant to time in language. It first discusses events and then times, covering for both the issues of definition, annotation and automatic processing.

2.2 Events

The Oxford English Dictionary defines an event as *"a thing that happens or takes place, especially one of importance"*. This definition could be broken down into occasions, actions, occurrences and states. However, the occasions, actions, occurrences and states are used in natural language more widely than this definition permits; there are often mentions of negated events, conditional events or modal events, which cannot be said to certainly "happen or take place" [7]. Further, events can be composed of many sub-events: for example, the Arab Spring lasted months and included multiple revolutions, each of which had a long history, a complex set of story threads all happening in parallel, a culmination and an aftermath. Indeed, processing historical events has its own challenges [8]. In addition, the definition of an event mention varies. Events may be represented by a variety of lengths of expressions, ranging from document collections [9] to single tokens. For the purpose of this book, the description of events from TimeML (a temporal markup language, [10]) is adopted, as follows:

> We consider "events" a cover term for situations that happen or occur. Events can be punctual or last for a period of time. We also consider as events those predicates describing states or circumstances in which something obtains or holds true.

Given that they may describe an action or transition, events are often expressed by verbs (*"The bus stopped suddenly"*). A nominalised event is an event that is represented by a noun phrase. For example, one might mention *the explosion*, which is a noun that describes an event. Events may also be expressed by statives, as in *the man was an idiot*); by predicatives, as in *Elizabeth is queen*; by adjectives, in *the storm is active*; and by prepositional phrases, such as in *soldiers will be present in uniform*. A further discussion of events and states can be found in [11].

Events do not have to be real and observable for them to be annotated in a given text. Unreal events, such as those in a fictional or modal context should be included in a temporal annotation of a document. Description of future events or of things subordinated into the conditional world of an *if* (for example) are still events, and ought to be processed as such.

2.2.1 Types of Event

Independent of their form of expression, events may be taxonomised into discrete classes. These are introduced as follows.

Occurrences

These denote something factual that happens or occurs. The event is not modal or intensional, and the account of the event is given first-hand. For example, *There was an explosion shortly before 11a.m.*.

Reports

These events are those of some actor relaying information about other events or states. The actor may be declaring, narrating, commenting upon or otherwise reporting. Typically in English, this class of events is expressed with words such as *said*, *told* and *explained*.

Perceptions

In some contrast to reports, perceptions are events that describe the observation or capture of some other event. Typical words that might be used for events in this class include *hear*, *see* and *discover*.

States

This class of events introduces something that holds true, such as an observation about world state.

Intensional Actions

These involve some actor with a specific (perhaps unstated) goal in mind, who performs distinct actions following that intent. The event is the expression of intentionality. Examples include *Microsoft tried to monopolize internet access.*

Aspectual

Finally, aspectual events are those expressions which describe certain parts of the life of an event, such as its beginning, culmination, continuation and so on. For example, *The scientists were starting to show signs of exhaustion*. See also [12].

While it is possible to sometimes further sub-categorise events, or group them in other ways, this coarse separation of event classes is ample for the scope of this book.

2.2.2 Schema for Event Annotation

Given definitions of events and a need to process them automatically, some kind of formal method of describing events must be introduced. For this, and for temporal annotation over the remainder of this book, we adopt TimeML. TimeML [10] is an XML-style markup for temporal information in natural language texts and has become an ISO standard. An overview of the syntax and annotation guidelines can be found online.[1]

[1] See http://www.timeml.org/.

TimeML proposes annotating events expressed in text with the <EVENT> tag, which has an class attribute. The class attribute contains one of a set range of values, depending on the class the event belongs to. TimeML's event class taxonomy is slightly richer than the one described above but essentially similar.

It is important to determine exactly what to annotate. Events may have actors, for example, and may be expressed using auxiliaries, prepositional phrases, negation and modal signifiers, and so on. The contiguous sequence of words that describes an event is called the event chunk. The single most important word within this chunk – the one that critically defines the event, such as the dominant verb – is the chunk (or event) head. In TimeML, <EVENT> annotations are applied to the shortest possible phrase that could describe the event; e.g., its head. See Example 2 from the TimeML 1.2.1 annotation guidelines.

Example 2 He would not have been going to permit anything like that.

In the example, negation, modality and an auxiliary-based tense structure are applied to the event, but only the head of the phrase is to be annotated.

TimeML also allows the annotation of extra information regarding events. This information may not be critical to the temporal significance of the event, but is certainly of linguistic interest and has proven helpful to many automatic annotation systems. The auxiliary attributes available are rough guidelines, rather than a precise or exhaustive set of temporal facets of events. Attributes of events annotated include:

- Part of speech (noun, verb etc.);
- Tense, from a limited set of values;
- Aspect, covering progressiveness and perfectiveness;
- Cardinality, indicating how many times the event may have been repeated;
- Polarity, to capture negation;
- Modality, holding the type of modality (if any) that applies to the event.

2.2.3 Automatic Event Annotation

2.2.3.1 Task Description

Complete event annotation comprises **event recognition** (determining which expressions denote events) and **event classification** (characterising events once found). Recognition concerns determining which words or phrases can be marked up as being events. Event classification involves determining the "class" of a particular event (such as an action or a state) according to a schema such as that presented in Sect. 2.2.1. Performing both tasks together is generally harder than just recognising where events lie in text [13].

2.2.3.2 Evaluation

In automatic event annotation, both recognition and classification of events need to be evaluated. Firstly, it should be possible to score a system's performance at identifying the textual extents of event words or phrases. Secondly, the assigned class of an event needs to be evaluated. This can be done with a simple correct or incorrect choice, leading to an overall accuracy score for a set of event class assignments.

Identifying Event Extents

Event recognition is the task of identifying and delimiting event phrases. A perfect system will mark all events, determining their textual bounds correctly and not mark any text that is not an event. Evaluation metrics should thus reward systems for both finding events and also for not finding non-events. Precision and recall fit these requirements and are often used to evaluate event recognition [14]. A brief description of precision and recall follows.

Recall is the proportion of existing items that have been identified by a system; a system that returns one event in a document that actually contains ten has a recall of 10 %. However, a system that marks everything as an event is bound to find all events and has a recall of 100 %. To balance this, one may introduce precision. Precision is the proportion of returned items that are correct; returning just one correct item and no others gives 100 % precision, but returning everything where there are only a few events will generate a low precision score.

Assuming events are always exactly one word long, if W is the set of identified words and E is the set of words that are events, we can define precision and recall as follows.

$$recall\ R = \frac{W \cap E}{W} \tag{2.1}$$

$$precision\ P = \frac{W \cap E}{E} \tag{2.2}$$

Relations between precision and recall are discussed by [15]. It is common to combine the two with a harmonic mean such as F-measure [16]. The formula is as follows:

$$F_\beta = (1 + \beta^2)\frac{PR}{\beta^2 P + R} \tag{2.3}$$

This is also known as the F1 score. The "1" in F1 corresponds to a weighting between precision and recall, with them being equal. A flexible F_β measure is also available, with low β favouring precision and high β favouring recall. A β of 0.5 may be desirable if one wants to particularly penalise spurious event annotations.

2.2.3.3 Approaches

Recognising and annotating event mentions in text has been approached in a variety of ways. It has been approach in a variety of ways, cast separately as a named entity recognition problem or as a syntactic analysis problem. The current most successful approaches combine both these approaches, and use semantic role information to reach comparatively high performance.

Boguraev et al. [13] cast TimeML EVENT recognition as a machine learning chunking problem. Text is treated as a sequence of tokens to which labels are assigned which describe chunk boundary information; three labels are possible – E for an end of a chunk, I for a token inside a chunk and O for "any token outside a target chunk". Features are then generated based on capitalisation, n-gram, part of speech, chunk type and head word information, similar to a word-profiling approach to entity recognition [17]. Following this, recognising EVENT extents in the Wall Street Journal is 77–80% accurate (F-measure). This figure drops to 61–64% accuracy for the joint task of recognising event extents and then correctly assigning TimeML classes to these events. The difference shows that the event classification task is non-trivial, having similar success rates to the approach used here for event recognition (e.g. around 75–80%).

EVITA [18], included in the TARSQI toolkit (Section A.3.1), employs different strategies for dealing with verb, noun and adjective events. It uses both machine learning and knowledge-based techniques. Verbs are filtered based on the verbal chunk head, modal auxiliaries and event polarity. Nouns are filtered against a look-up table and sense disambiguation lookup (to repeat the example from the paper, a noun in WordNet synset *phenomenon* is not an event if is it also subsumed by the synset *cloud*). Finally, adjectives are only tagged as events if they have already been used as such by a gold standard source (such as TimeBank). EVITA reaches 80 % F-measure when recognising verbal events in TimeBank 1.2, which is comparable to IAA scores from that corpus' creation.

More recent efforts in automatic TimeML event annotation focus on machine learning approaches incorporating information about semantic roles, reaching F-measures of over 0.80. One leading tool, TIPSem-B [19], incorporates semantic role information into its CRF-based event annotation approach. It is openly available for download.[2] Other approaches have refreshed existing systems like EVITA and included the whole into common NLP frameworks; for example, GATE-Time [20] adapts EVITA into a machine learning system as a GATE component, making it easy to port between applications and capable of using extra training data to improve performance.

[2] See http://gplsi.dlsi.ua.es/demos/TIMEE/.

2.3 Temporal Expressions

Temporal information in text is often expressed using a phrase that precisely describes a point or duration. Sometimes these points reference an absolute unambiguous time (anchored via e.g. a calendar), which is of great help when trying to map events from a discourse to a timeline. It is also often that case that such phrases explicitly state an interval's length. Because they are so explicit, these phrases are used when temporality is critical. Thus, attempts to extract a discourse's temporal information must capture and process these phrases.

Linguistic characterisation of temporal expressions has led to discussion and observations regarding their usage and situation. Hitzeman [21] found that time expressions are often used as discourse segmentation markers and highlights their potential ambiguity. They find that the interpretation of a given temporal expression depends on its syntactic position. Similarly, Bestgen and Vonk [22] show that temporal expressions used as adverbials help set the scene for a sub-part of discourse, providing a context and a timeframe and are helpful discourse segmentation markers, improving discourse comprehension. Cohen and Schwer [23] perform multi-lingual characterisation of temporal markers, describing such expressions as comprising three parts: the size of the temporal segment, the distance from a temporal centre (e.g. a reference point, a concept addressed in detail during Sect. 6.3.1) and an orientation such as future or past. Finally, [24] is entirely dedicated to temporal expressions and the current reference book on the topic.

For this book, a "temporal expression", or **timex**, is any expression that denotes a moment, interval or other temporal region without having to rely upon an event. Each interval is composed of two points between which it obtains. For example, *24th August 1997*, *two weeks* and *now* are all temporal expressions; *after the storm* is not. Hobbs and Pan [25] define a "proper interval" as one where the start point is before the end point. Under this definition, this book considers only "proper interval" as intervals; that is, no minimum atomic duration is recognised, and there is no quantisation of time into chronons. Rather, temporal entities are described by infinitesimal points that bound them.

One needs to discover where these expressions occur in text and understand something of their semantics before being able to connect them using temporal relations.

2.3.1 Temporal Expression Types

Before describing algorithms that can identify and anchor time expressions, we will briefly equip the reader with a short summary of types of time expression. Most papers that cover this topic, using varying nomenclatures, settle on a small set of different types of time expressions defined by their authors [26–30]. These types can generally be mapped onto one of the following distinct classes.

- **Absolute** — Where the text explicitly states an unambiguous time. Depending on the granularity of the interval, the text includes enough information to narrow a point or interval directly down to one single occurrence. This is in contrast to a time which, while precise and maybe easy for humans to pin onto a calendar, relies on an external reference. For example, the interval *Thursday October 1st, 2009* would be considered absolute, but *The week after next* would not – the information is not all explicit or held in the same place; this latter expression implies reliance on some external reference time. Absolute expressions are sometimes also known as fully-qualified time expressions.
- **Deictic** — Cases where, given a known time of utterance, one can determine the period being referred to. These time expressions, specify a temporal distance and direction from the utterance time. One might see a magazine bulletin begin *Two weeks ago, we were still in Saigon.*; this expression leaves an unclear implicit speech time, which one could safely assume was the date the article was written. More common examples include *tomorrow* and *yesterday*, which are both offset from speech time; to describe this using Reichenbach's model (Sect. 6.3), deictic temporal expressions represent situations where reference time and speech time are the same.
- **Anaphoric** — Where speech and reference time are not at the same point. Anaphoric temporal expressions have three parts – temporal distance (e.g. 4 days), temporal direction (past or future) and an anchor that the distance and direction are applied from. The anchor, for anaphoric temporal expressions, is the current reference time as per Reichenbach's model (Sect. 6.3). Example phrases include *the next week*, *that evening* or *a few hours later*, none of which can be anchored even when their speech time is known.
- **Duration** — A duration describes an interval bounded by a start and an end, where the distance between the two is known, but the expression itself is not placeable on any external time system (like a calendar). Durations generally include a time unit as their head token; for example, *ninety minutes* is a single duration timex. This type of temporal expression is easily confused with deictic expressions; to use Ahn's example [28],

Example 3 "In the sentence *The Texas Seven hid out there for three weeks*, the timex *three weeks* refers to a duration, whereas in the sentence *California may run out of cash in three weeks*, the same timex refers to a point three weeks after the reference point".

- **Set** — Regularly recurring times, such as "every Christmas" or "each Tuesday". These usually have a regular interval between occurrences and persist for a duration or describe a point event ("every other Thursday at 4.30pm").

2.3.2 Schema for Timex Annotation

Temporal expressions are often inherently vague, and typically only communicated only to the level of precision that the speaker requires in order to convey their point coherently. As a result, it is difficult to develop a precise, discrete knowledge representation form for timexes – the classic AI problem of building machine-readable forms from qualitative concepts. Bearing this in mind, approaches to timex annotation have been developed.

Direct anchoring points for times and events comprise normalised temporal expressions – that is, linguistic expressions that refer to a time, which can be placed onto an absolute calendar scale. For example, *"2 July 2009"* is an unambiguous date. Some reasoning may be required in order to normalise a temporal expression; one may encounter text such as *"on Sunday"*, which requires a reference temporal expression that is better specified before it can be absolutely positioned. The recognition, categorisation and normalisation of temporal expressions is briefly discussed in Sect. 2.3.

To this end, any timex annotation schema has to account for describing both the extents of the expression and the value of the expression itself. Today, the two prevailing standards for timex annotation are TIMEX2 and TIMEX3. These standards evolved through the MUC [31] exercises and TERN [13] through TIMEX to more recent incarnations. Both are XML-based and cater for the timex classes of duration, time, date and set.

An annotation schema should provide a way of marking up events, times and relations in text. Additional information can be provided, such as normalisations of times, tense and aspect information, markup of temporal signals such time adverbials, aspectual links and so on. This book works with the TimeML annotation standard, as it is the most active and has the largest amount of annotated resources. TimeML accounts for not only timexes but also event and temporal relation annotation. Only the timex aspects are discussed in this section.

This section introduces the TimeML, TIMEX and TCNL annotation schemas. Other notations are available, but as the future work in this book concentrates on TimeML, an exhaustive cataloguing would not be appropriate.

2.3.2.1 TimeML

TimeML [10] is an XML-based language for temporal annotation. It allows annotation of events and times, with a rich format for each, as well as thorough provision of links to capture relations between events and times:

- TLINK: temporal, possibly including references to supporting words
- SLINK: subordinate, for modality, evidentials and factives
- ALINK: aspectual, only between two events, describing an aspectual connection

As well as this, TimeML includes a comprehensive event annotation and uses the TIMEX3 standard described above for representing temporal expressions. One may also link signals (such as temporal adverbials) with events or temporal links, to show sources of temporal information in text. TimeML is the only temporal annotation language to become an ISO standard.[3] Widespread adoption has lead to many temporal information extraction experiments using TimeML annotated corpora, as well as multiple iterations of the language and the production of processing tools that can parse the markup.

TimeML does not employ the Allen interval relations, but instead uses its own set, based on Allen's earlier work [32, 33]. Notably, TimeML has no OVERLAPS relation, or way of expressing it. This is clarified in TimeML-strict [34]. A fuller introduction to TimeML can be found in [35].

ISO-TimeML [36] is a LAF and TEI compatible iteration of TimeML. It permits stand-off annotation, where the SGML annotations do not clutter text by being inline and has a more elegant method of instantiating events. The formal standard is recognised by the ISO and maintained by an active working committee.

2.3.2.2 TIMEX3

TIMEX3 stipulates the annotation of smaller strings than TIMEX2 [27] and is intended for use alongside mechanisms for annotating temporal links and events. TIMEX2 permits longer expressions, including event-based timexes which are anchored not to absolute scales but to events described in the text; the rationale behind this is that TIMEX2 was not designed for use in an environment that included event annotations, whereas TIMEX3 is intended to be used as part of the TimeML annotation scheme. TIMEX3 is designed to work across domains [37]. Focused research further details the differences between the two standards and describes an approach for converting data from the TIMEX2 to TIMEX3 standard [30].

TIMEX3 is currently used as the means of describing times in TimeML; it looks like this:

```
<TIMEX3
        tid="t43" type="DATE"
        value="1989-10-30"
        functionInDocument="CREATION_TIME">
                10/30/89
</TIMEX3>
```

The `value` field may take the form an ISO8601-format date, a `P` followed by a numeric quantity and unit symbol to denote a period, or one of a number of special anaphoric-based values such as `PRESENT_REF`. Its format is not trivial and the TIDES/TimeML documentation are the best resources for its description [10, 27]. For the scope of this book, we generally consider TIMEX3 in the context of

[3] ISO WD 24617-1:2007

TimeML, as we are interested in an annotation schema that covers not only temporal expressions but also events and temporal relations. Other attributes of TIMEX3 annotations include:

- Function in document, to denote special timexes such as the document creation point;
- Type, to capture the timex class;
- Modifier for adding information that cannot be added to the value, such as qualitative information (e.g. *"the dawn of 2000"* would be marked as the year 2000 with a modifier of `start`);
- Quantifier and frequency for describing the repetition pattern of a set timex; for example, *"every other Sunday"* would have a value of `P1W` (period of 1 week) and a quantifier of `every other` – and *"twice a day"* has value `P1D` with a frequency of 2.

2.3.2.3 TCNL

TCNL [29] is "a compact representational language" for working with time expressions. A set of operators and labels are defined, which can be combined to produce various offsets or absolute expressions. For example, TCNL looks elegant for simplistic temporal relations; $\{tue,\ <\ \{|25\{day\}|@\{dec\}\}\}$ for *Tuesday before Christmas*, or $\{friday,\ <\ now\}$ to represent an earlier Friday. A calendar model, working with different levels of granularity, is used to help anchor times. Weeks and months, for example, have different durations and do not share synchronised boundaries, but both – when combined with an integer – can define a solidly bounded absolute interval; e.g. Week 34 2008, or January 2012.

Its authors suggest that TCNL has benefits over TOP [38], TIMEX2 and TIMEX3/ TimeML; namely, that TCNL is calendar-agnostic, focuses on intensional meaning of expressions (which are allowed in TimeML, but not compulsory and not used in the two largest TimeML corpora), shows contextual dependency by using references such as focus and that its type system makes granularity conversion transparent.

An example of TCNL's capture of intensionsal time reference – "Yesterday" becomes $\{now - |1day|\}$ instead of something like 20090506. A set of operators are used to reason between operands:

- $+/-$ for forward/reverse shifting.
- @ for in; e.g., $\{|2sun|@\{may\}\}$ is "the second Sunday in May".
- & for distribution; e.g., $\{15hour\}\&[\{wed\} : \{fri\}]\}$ is "3pm from Wednesday to Friday".

Performing some basic algebra, "Friday last week" is split, into "Friday" and "last week". This is represented thus:

$\{fri\} + \{now - |1week|\} = \{fri, \{now - |1week|\}\} = \{now - |1fri|\}$

Further examples in TCNL and a reference guide to the language, can be found in [39].

2.3.3 Automatic Timex Annotation

2.3.3.1 Task Description

As with events, extracting timexes can be decomposed into a multi-part task. In this case, the principal parts are determining which words and phrases in a document comprise timexes and then assigning various attribute values to that phrase. Once identified, a temporal expression may be converted to a fully specified date or interval. Existing work has investigated the task of "anchoring" or "normalising" temporal expressions; that is, taking tokens from a document and mapping them to an absolute temporal scale, using an agreed notation. For example, while the single 24-hour period that the expression *next Thursday* refers may be immediately clear to us at any given time, some processing is required on the part of a computer to map this into a complete time (specifying at least a year and day). It is also important to choose the correct granularity for temporal expressions; *next day* refers loosely to the contents of a 24-hour period, not to a period of precisely 86400 seconds occurring between two local midnights (or however many caesium decay events, in SI terms).

Automatic timex annotation is typically a three-stage process. Firstly, one must determine the extents of a temporal expression. This stage may be evaluated using conventional precision and recall measures. Secondly, the timex should be interpreted [40], converting it to a representation according to an established convention. This includes assigning both an expression type and value, which can be evaluated with string matching for strict evaluation. Thirdly and optionally, the timex may be anchored to a time scale, which involves mapping it to a specific time, date, or range of times and dates.

Even in the case of temporal expressions, apart from those that are absolutely anchored in text – that is, those that include a year placed along an agreed calendar system – one will have to use some knowledge to normalise an expression, based on other information. One cannot determine precisely which "2 July" is referred to without a contextual clue of the year. These clues may be from the document creation time, or from a recently specified absolute temporal expression which sets reference time (see Sect. 6.3); failing that, the information again has to come from relations between temporal expressions.

2.3.3.2 Evaluating Temporal Expression Annotation

Precision and recall are suitable for evaluating temporal expression recognition (see Sect. 2.2.3.2). Temporal expressions can also be broken down into one of many classes and may be interpreted or even anchored to a calendar. To evaluate temporal expression typing, a simple "proportion correct" or accuracy metric works well. Interpretation and anchoring efforts can be compared verbatim to a gold standard to assess accuracy. One must also choose whether or not to allow equivalent matches to be considered as equal. For example, the TIMEX3 values `P1D` and `P24H` both correspond to a duration of a day and may be considered equivalent. However, if one prefers an annotation that matches the exact language used in a document, it may be argued that *"one day"* should only be given a value of `P1D` and that `P24H` is more representative of text like *"24 hours"*. Any timex evaluation needs to take a stance on these issues.

2.3.3.3 Timex Annotation Systems

Rule based systems are frequently employed in approaches to these tasks, because plenty of set phrases are used to describe time and they employ a simple grammar. In fact, one very successful approach to normalising week days is entirely rule-based [41]. This attribute of temporal expressions means that finite state grammars can be used for Timex recognition [13, 17]. In the case of the first paper, a rule-based system was completed by interleaving finite state grammars with named entity recognition, in order to enable temporal expressions in linguistic units, as opposed to lexical ones. This enables the identification of events and associations that are semantically present in a sentence but not immediately obvious from its construction.

Some systems, such as GUTime [26], rely heavily on a rule-based approach to spotting sequences of tokens, as there are many temporal expressions present in the English language that can be identified and anchored in this way. Named entity recognition (NER) has also been used to identify times in text [42].

Following MUC6, MUC7, TERN and ACE, TempEval-2 also included a task for temporal expression annotation. The entered systems and subsequent improvements have provided clear advantages over prior attempts in temporal expression annotation. Because timex annotation is not the primary focus of this book, only TempEval-2 and later experiments are described here.

In this task, rule-based, machine learning-based and hybrid systems all performed well at timex recognition. For English the timex extent recognition performance F-measure ranged from 0.26 to 0.86, with an average of 0.78. The best performance was with F1 of 0.86; seven systems reached F-measures of 0.84-0.86. This is promising, though by no means a solution to the timex recognition problem. Timex classification was performed best by a TIMEX2 transduction system with accuracy 0.98 [43], though all but two systems attempting timex classification reached at least 90 % accuracy.

Normalisation proved to be a substantially harder task, results ranging from 0.17 to 0.85. This task can involve complex reasoning and demands large and diverse amounts of training data [44]. The number of possible values is high, so giving default answers (e.g. most-common-class values) as a back-up is unlikely to be of any use.

Three systems in particular worked best at TempEval-2, though their strengths lie in different places. HeidelTime [45] is a modular rule-based system including a large ruleset; this enabled it to achieve top performance at timex normalisation, and has been used to power such applications as temporal information retrieval [46, 47]. However, rule-based approaches are likely to face diminishing returns as they attempt to raise recall through introduction of new rules [48]. TRIPS/TRIOS [49] and TIPSem-B [50] are both systems that use machine learning for timex recognition, with sophisticated feature sets. Using the TempEval-2 data released after the exercise, it has been shown to be possible for very simple feature sets to reproduce state-of-the-art timex recognition performance [51]. Normalisation remains a task that appears to requires a rule-driven solution, with promising new systems emerging [52].

2.4 Chapter Summary

This chapter has introduced the concepts of a timex and an event, and given formal definitions and annotation schemas for them, as well as describing the state of the art in their automatic annotation. We consider events and times as being anchored to a minimal representation in a document, typically a single word for events and a few words for temporal expressions (timexes). Conceptually, they are modelled as temporal *intervals*, having both a start and end instant, and holding for the period between. Events and times are the foundational building blocks of temporal discourse annotation, and both are considered as intervals whenever possible. The following chapter will cover the next step: temporal relations between intervals.

References

1. Moens, M., Steedman, M.: Temporal ontology and temporal reference. Comput. Linguist. **14**(2), 15–28 (1988)
2. Pan, F., Mulkar, R., Hobbs, J.: Learning event durations from event descriptions. In: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, pp. 393–400. Association for Computational Linguistics (2006)
3. Gusev, A., Chambers, N., D.R., K., Khaitan, P., Bethard, S., Jurafsky, D.: Using query pattens to learn the duration of events. In: Proceedings of the 9th International Conference on Computational Semantics, pp. 145–154 (2011)
4. Mani, I., Schiffman, B.: Temporally anchoring and ordering events in news. Time and Event Recognition in Natural Language (2005)
5. Mani, I., Wellner, B., Verhagen, M., Pustejovsky, J.: Three approaches to learning TLINKS in TimeML. Tech. Rep. CS-07-268, Brandeis University, Waltham, MA, USA (2007)
6. Bethard, S., Martin, J., Klingenstein, S.: Timelines from Text: Identification of Syntactic Temporal Relations. In: Proceedings of the International Conference on Semantic Computing, pp. 11–18 (2007)
7. Pustejovsky, J.: The syntax of event structure. Cognition **41**(1–3), 47 (1991)
8. Cybulska, A., Vossen, P.: Historical event extraction from text. In: Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities, pp. 39–43. Association for Computational Linguistics (2011)
9. Ritter, A., Etzioni, O., Clark, S., et al.: Open domain event extraction from Twitter. In: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 1104–1112. ACM (2012)

10. Pustejovsky, J., Ingria, B., Sauri, R., Castano, J., Littman, J., Gaizauskas, R.: The Specification Language TimeML. In: The Language of Time: A Reader, pp. 545–557. Oxford University Press (2004)
11. Steedman, M.: Reference to past time, pp. 125–157. Speech, Place and Action (1982)
12. Vendler, Z.: Verbs and Times. The philosophical review. Ithaca, New York (1957)
13. Boguraev, B., Ando, R.: TimeML-compliant text analysis for temporal reasoning. In: Proceedings of International Joint Conference on Artificial Intelligence (IJCAI) (2005)
14. Verhagen, M., Saurí, R., Caselli, T., Pustejovsky, J.: SemEval-2010 task 13: TempEval-2. In: Proceedings of the 5th International Workshop on Semantic Evaluation, pp. 57–62. Association for Computational Linguistics (2010)
15. Buckland, M., Gey, F.: The relationship between recall and precision. J. Am. Soc. Inf. Sci. **45**(1), 12–19 (1994)
16. van Rijsbergen, C.: Information retrieval, 2nd ed. Butterworths (1979)
17. Ando, R.: Exploiting unannotated corpora for tagging and chunking. In: Proceedings of the ACL interactive poster and demonstration sessions. Association for Computational Linguistics (2004)
18. Saurí, R., Knippen, R., Verhagen, M., Pustejovsky, J.: Evita: a robust event recognizer for QA systems. In: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, p. 707. Association for Computational Linguistics (2005)
19. Llorens, H.: A semantic approach to temporal information processing. Ph.D. thesis, University of Alicante (2011)
20. Derczynski, L., Strötgen, J., Maynard, D., Greenwood, M.A., Jung, M.: GATE-Time: Extraction of Temporal Expressions and Event. In: Proceedings of the 10th Language Resources and Evaluation Conference, pp. 3702–3708 (2016)
21. Hitzeman, J.: Semantic partition and the ambiguity of sentences containing temporal adverbials. Nat. Lang. Semant. **5**(2), 87–100 (1997)
22. Bestgen, Y., Vonk, W.: Temporal adverbials as segmentation markers in discourse comprehension. J. Mem. Lang. **42**(1), 74–87 (1999)
23. Cohen, D., Schwer, S.: Proximal deixis with calendar terms: Cross-linguistic patterns of temporal reference. ms. submitted in lingua (2012)
24. Strötgen, J., Gertz, M.: Domain-Sensitive Temporal Tagging. Morgan-Claypool (2016)
25. Hobbs, J.R., Pan, F.: An ontology of time for the semantic web. ACM Trans. Asian Lang. Inf. Process. (TALIP) **3**(1), 66–85 (2004)
26. Mani, I., Wilson, G.: Robust temporal processing of news. In: Proceedings of the 38th Annual Meeting on Association for Computational Linguistics, pp. 69–76. Association for Computational Linguistics (2000)
27. Ferro, L., Gerber, L., Mani, I., Sundheim, B., Wilson, G.: Tides 2005 standard for the annotation of temporal expressions. Tech. Rep. 03-1046, The MITRE Corporation (2005)
28. Ahn, D., Adafre, S., Rijke, M.: Towards task-based temporal extraction and recognition. In: Dagstuhl Seminar Proceedings, vol. 5151 (2005)
29. Han, B., Gates, D., Levin, L.: From language to time: A temporal expression anchorer. In: Proceedings of the 13th International Symposium on Temporal Representation and Reasoning (TIME) (2006)
30. Derczynski, L., Llorens, H., Saquete, E.: Massively increasing TIMEX3 resources: a transduction approach. In: Proceedings of the Language Resources and Evaluation Conference (2012)
31. Chinchor, N., Robinson, P.: MUC-7 named entity task definition. In: Proceedings of the 7th Message Understanding Conference (1997)
32. Allen, J.: Maintaining knowledge about temporal intervals. Commun. ACM **26**(11), 832–843 (1983)
33. Allen, J.F.: Towards a general theory of action and time. Artif. Intell. **23**(2), 123–154 (1984)
34. Derczynski, L., Llorens, H., UzZaman, N.: TimeML-strict: clarifying temporal annotation. arXiv preprint arXiv:1304.7289 (2013)

35. Pustejovsky, J., Knippen, R., Littman, J., Saurí, R.: Temporal and Event Information in Natural Language Text. Lang. Resour. Eval. **39**(2), 123–164 (2005)
36. Pustejovsky, J., Lee, K., Bunt, H., Romary, L.: ISO-TimeML: An International Standard for Semantic Annotation. In: Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10) (2010)
37. Strötgen, J., Gertz, M.: Domain-sensitive temporal tagging. Synth. Lect. Hum. Lang. Technol. **9**(3), 1–151 (2016)
38. Androutsopoulos, I.: Temporal meaning representations in a natural language front-end. arXiv preprint cs/9906020 (1999)
39. Han, B.: Reasoning about a Temporal Scenario in Natural Language. In: Proceedings of the International Joint Conference on Artificial Intelligence (2009)
40. Mazur, P., Dale, R.: LTIMEX: Representing the Local Semantics of Temporal Expressions. In: Proceedings of the 1st International Workshop on Advances in Semantic Information Retrieval (ASIR), pp. 201–208 (2011)
41. Mazur, P., Dale, R.: Whats the date? High accuracy interpretation of weekday. In: 22nd International Conference on Computational Linguistics (Coling 2008), Manchester, UK, pp. 553–560 (2008)
42. Grover, C., Tobin, R., Alex, B., Byrne, K.: Edinburgh-LTG: TempEval-2 system description. In: Proceedings of the 5th International Workshop on Semantic Evaluation, pp. 333–336. Association for Computational Linguistics (2010)
43. Saquete, E.: TERSEO+T2T3 Transducer: a systems for recognizing and normalizing TIMEX3. In: Proceedings of the 5th International Workshop on Semantic Evaluation, pp. 317–320. Association for Computational Linguistics (2010)
44. Brucato, M., Derczynski, L., Llorens, H., Bontcheva, K., Jensen, C.S.: Recognising and interpreting named temporal expressions. In: Proc. RANLP, pp. 113–121 (2013)
45. Strötgen, J., Gertz, M.: HeidelTime: High quality rule-based extraction and normalization of temporal expressions. In: Proceedings of the 5th International Workshop on Semantic Evaluation, pp. 321–324. Association for Computational Linguistics (2010)
46. Strötgen, J.: Domain-sensitive temporal tagging for event-centric information retrieval. Ph.D. thesis, Heidelberg University (2015)
47. Derczynski, L., Strötgen, J., Campos, R., Alonso, O.: Time and information retrieval: Introduction to the special issue. Information Processing & Management (2015)
48. Bethard, S.: A synchronous context free grammar for time normalization. In: Proceedings of EMNLP, pp. 821–826. Association for Computational Linguistics (2013)
49. UzZaman, N., Allen, J.: TRIPS and TRIOS system for TempEval-2: Extracting temporal information from text. In: Proceedings of the 5th International Workshop on Semantic Evaluation, pp. 276–283. Association for Computational Linguistics (2010)
50. Llorens, H., Saquete, E., Navarro, B.: TIPSem (English and Spanish): Evaluating CRFs and Semantic Roles in TempEval-2. In: Proceedings of SemEval-2010, pp. 284–291. ACL (2010)
51. Llorens, H., Saquete, E., Navarro, B.: Syntax-motivated context windows of morpho-lexical features for recognizing time and event expressions in natural language. Nat. Lang. Process. Inf. Syst. **6716**, 295–299 (2011)
52. Llorens, H., Derczynski, L., Saquete, E., Gaizauskas, R.: TIMEN: An Open Temporal Expression Normalization Resource. In: Proceedings of the Language Resources and Evaluation Conference (2012)

Springer

http://www.springer.com/978-3-319-47240-9

Automatically Ordering Events and Times in Text
Derczynski, L.R.A.
2017, XXI, 205 p. 25 illus., Hardcover
ISBN: 978-3-319-47240-9